AD-A052 359 RAYTHEON CO WAYLAND MASS ADVANCED DEVELOPMENT LAB F/G 17/9
R AND D EQUIPMENT INFORMATION REPORT: TRACKING AND SIGNIFICANCE--ETC(U)
NOV 77 T I BOAK, A J JAGODNIK, R B MARSHALL F19628-77-C-0148
UNCLASSIFIED ER77-4283 AFGL-TR-77-0259 NL

1 OF 2
AD
A052359

2

AFGL-TR-77-0259

R&D EQUIPMENT INFORMATION REPORT

Tracking and Significance Estimator

T.I.S. Boak, III
A.J. Jagodnik, Jr.
R.B. Marshall
D. Riceman
M.J. Young

Raytheon Company
Equipment Division
Advanced Development Laboratory
Boston Post Road
Wayland, Massachusetts 01778

7 November 1977

Final Report
77 April - 77 October

Approved for public release; distribution unlimited.

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9 Final rept. Apr-Oct 77,

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18 AFGL-TR-77-Ø259	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 R&D Equipment Information Report: and Tracking and Significance Estimator.		5. TYPE OF REPORT & PERIOD COVERED Final Report 77 April-77 October
		6. PERFORMING ORG. REPORT NUMBER 14 ER77-4283
7. AUTHOR(s) 10 T.I. Boak, III, A.J. Jagodnik, Jr., R.B. Marshall, D. Riceman, M.J. Young		8. CONTRACT OR GRANT NUMBER(s) 15 F19628-77-C-Ø148 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Raytheon Company Equipment Division Advanced Development Laboratory Wayland, MA 01778		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 16 667Ø0201 17 Ø2
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory Hanscom AFB, Massachusetts 01731 Monitor/Alexander W. Bishop/LYW		12. REPORT DATE 11 7 Nov[redacted]977
		13. NUMBER OF PAGES 118 12 119p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Meteorological Radar Data Processing, Storm Cell Tracking, Operational Weather Radar Data Display

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Equipment is described which operates on data obtained by a coherent meteorological radar with doppler pulse pair processor. The Tracking and Significance Estimator (TSE) periodically acquires logarithmic power, mean doppler velocity, and variance of doppler velocity for an entire azimuthal antenna scan. The TSE finds the twelve most severe storm cells, computes their reflectivity-weighted centers, then plots the centers and the cell outlines on a color CRT display. Centers from up to 24 previous acquisition intervals, color coded according to acquisition time, are also presented so as to form

298 353 JOB

storm cell tracks.

TABLE OF CONTENTS

Section Page

1. INTRODUCTION 6

2. GENERAL DESCRIPTION 7
2.1 System Configuration 7
2.2 Operational Capabilities 10
2.2.1 Data Collection 11
2.2.2 Storm Cell Processing 14

3. OPERATION OF TSE 16
3.1 Start Up 16
3.1.1 Power Up 16
3.1.2 Re-Loading the OS/32-MT System 17
3.1.3 Initialization To Run TSE 17
3.1.4 Running DISCHECK 18
3.2 Execution of Tracking and Significance Estimator 18
3.2.1 TSE Command 19
3.2.2 TCO Command 22
3.2.3 TPA Command 22
3.2.4 DUMP Command 22
3.3 Terminating TSE 23
3.4 Powering Down the 7/32 23

4. DETAILED HARDWARE DESCRIPTION 24
4.1 New Equipment 24
4.1.1 Pulse Pair Recorder Interface 25
4.1.2 Scan Converter Refresh Memory Interface 36
4.2 Existing Equipment 37
4.2.1 PPP Recorder 37
4.2.2 Scan Converter Refresh Memory 37

TABLE OF CONTENTS
(Continued)

Section		Page
5.	DETAILED SOFTWARE DESCRIPTION	41
5.1	Support Software	41
5.1.1	Operating System	41
5.1.2	Drivers	42
5.2	Task Description	49
5.2.1	TSEPARAM.CAL	49
5.2.2	TSEMAIN.FTN	50
5.2.3	TSEHIST.FTN	51
5.2.4	TSEDATA.FTN	51
5.2.5	TSERANGE.FTN	53
5.2.6	TSEAZPRO.FTN	55
5.2.7	TSECENT.FTN	56
5.2.8	TSEPLT.FTN	57
5.2.9	TSEDUMP.FTN	57
5.2.10	TSECHNG.CAL	58
	APPENDIX	
A	TEST SOFTWARE FOR PPRI	60
B	SCRMI TEST SOFTWARE (SCRMITST)	65
C	COMMON UTILIZATION IN TSE	69
D	ECHO TRACKING AND SIGNIFICANCE ESTIMATOR PHOTOGRAPHS	74
E	LIST OF SCHEMATICS	81
F	INTERDATA PRODUCT BULLETINS	83
G	A.J. JAGODNIK, JR. MEMO #AJJ-76A "CURSOR DATA TRANSFER IN THE DISPLAY DATA INTERFACE"	116

LIST OF ILLUSTRATIONS

Figures		Page
2-1	TSE Block Diagram	8
2-2	TSE System Timing	10
4-1	Encoder Output Formats Showing Re-Ordering of Bits Into Interdata Fullwords	27
4-2	Interdata Full Words for Video and Ancillary Data Entered Via DMA Through PPRI from PPP Encoder	28
4-3	PPRI Block Diagram	30
4-4	Timing Diagram for Initialization and Startup of PPRI	34
4-5	Timing Diagram Showing FIFO Timing in PPRI	35
5-1	Elements of I/O System	41
5-2	Buffer Format for SCRM Driver	48
A-1	Flow Chart for PPRI Test Program	63
D-1	Interdata 7/32 Processor and 10 M Byte Disc Drive	75
D-2	Line Printer, Card Reader, and System Console	76
D-3	Scan Converter/Refresh Memory	77
D-4	Pulse Pair Processor, Encoder/Decoder (Right) and Tape Recorder (Left)	78
D-5	Reflectivity Data	79
D-6	TSE Output on Data in Figure D-5	80

SECTION 1. INTRODUCTION

Doppler weather radars are capable of sampling vast quantities of data such as reflectivity or doppler velocity spectra which are related to variables of interest to the meteorologist. The Pulse Pair Processor (PPP), developed by Raytheon for AFGL under contract F19628-72-C-0293, together with the recorder system added later under contract F19628-74-C-0079, have made progress toward reducing these data to a form suitable for radar meteorological research and archiving them for future study. In particular, the PPP develops logarithmic reflectivity, mean doppler velocity, and doppler spectrum variance for 256, 512, 768, or 1024 range cells of 0.5, 1, or 2 microseconds (75, 150 or 300 meters) with batch-type integration of between one and 1024 radar pulse periods. The amount of information generated in a single 360 degree azimuth scan, typically less than a minute in duration, approaches 10 million bits with a one-degree-beamwidth antenna.

While the fine detail available in the PPP or recorder outputs is well matched to the requirements of the research radar meteorologist interested in analyzing the small-scale structure of weather returns, the more operationally-oriented meteorologist might well despair at being presented with 10 million bits of information per minute. What is needed is further real-time processing which automatically makes sense out of the vast quantity of information available at the PPP outputs, especially during critical severe storm conditions. The Echo Tracking and Significance Estimator (TSE) will perform such real-time processing with the objective of providing a simplified estimate of the important features of each storm cell, as well as its time-history.

SECTION 2. GENERAL DESCRIPTION

In this section, the TSE is described in general terms: first, as a collection of hardware; then functionally in terms of operational capabilities of the system.

2.1 System Configuration

The TSE is illustrated in block diagram form in Figure 2-1, where it can be seen that the heart of the system is an Interdata 7/32 Processor with 192K Bytes of core memory. The OS/32 disc-based multi-tasking operating system occupies about 70K Bytes of this store; thus leaving ample capacity for processing programs and data buffers as well as for development programs such as editors, assemblers and compilers. Low and medium speed peripherals communicate with the CPU through the I/O MULTIPLEXER BUS. Either the CRT terminal or the Carousel can be configured as the command console for the system. The Card Reader and Line Printer are normally used as the principal program input and output-listing devices. The Loader Storage Unit (LSU) simplifies initial loading of the operating system from disc while the Universal Clock Module provides interrupts at programmable time intervals.

A Scan Converter Refresh Memory Interface (SCRMI), constructed on a Universal Logic Interface Module (ULIM), permits use of one or more of the SCRM's bit-image memory/color display channels for plotting outputs in color-graphic form. This ULIM has a second channel which could someday accommodate an antenna scan controller.

The Magnetic Disc bulk storage unit has a 5 Megabyte two-surface fixed platter and a 5 Megabyte removable cartridge, also with two recording surfaces. The four heads are activated by a common servo system which can reach any of the 408 4-track cylinders in 60 milliseconds or less. The 2400 RPM rotational rate, with 6144 bytes/track (formatted), results in a

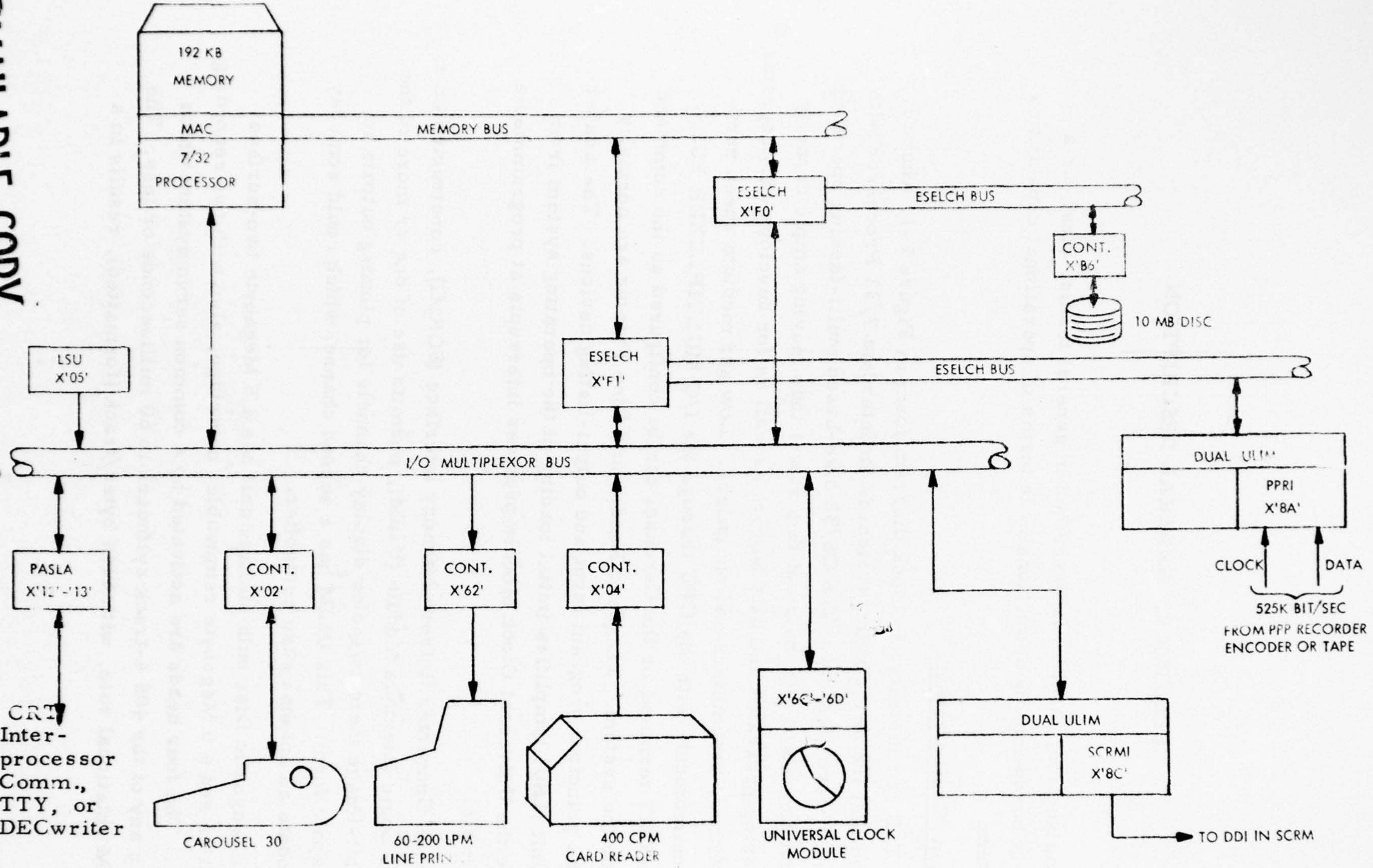


Figure 2-1. TSE Block Diagram

312,500 byte/second transfer rate which necessitates that the disc controller communicate with the processor via an Extended SELector CHannel. This ESELCH provides direct memory access by stealing one-microsecond memory cycles from the 7/32 while that processor executes other instructions.

Another ESELCH supports the major radar data input port of the TSE: the Pulse Pair Recorder Interface (PPRI), built on a ULIM as is the SCRMI. The PPRI has only two input lines: a 525K bit/sec serial data line and its corresponding bit clock. The serial data, originating either in real time at the PPP Encoder output or played-back from tape in the FR 2000A wideband instrumentation recorder, contains all radar information needed by the TSE. Included are both radar video data--mean and variance of the doppler velocity spectrum and power in each range cell, and ancillary data--antenna angles, time code, pulse pair processor status, PRF, etc. The PPRI performs the functions of serial-to-parallel conversion, sync code recognition, complementing of certain bits, conversion of some video data from sign-magnitude to two's complement, re-ordering of data bits into an Interdata-compatible 32-bit word, data buffering, and synchronization to accomplish transfer of these words into the processor by means of a DMA transfer through the 16-bit wide ESELCH.

The transfer of information takes place in blocks of 16 bytes of ancillary data followed by 1024, 2048, 3072, or 4096 bytes of radar data, depending on the number of range cells N_{RC} selected on the PPP. Each such block begins with an interrupt to the 7/32 and involves a transfer rate of 70K bytes per second, much lower than the disc rate but still high enough to justify the PPRI having its own ESELCH to eliminate the need for processor intervention during each block transfer.

Like the SCRMI, the PPRI only uses one of the two controllers available on the ULIM so that there is provision for future expansion--for example, the addition of a DMA channel for interprocessor communication.

2.2 Operational Capabilities

A good overall picture of TSE operation is most easily achieved by reference to the simplified timing diagram in Figure 2-2. The Radar antenna scans in azimuth with a period of 50 seconds in this example. The Data Collection process begins with the transfer of PPP/Radar data to the disc; this process continues until an entire scan has been collected.

After the Data Collection task, the Range-Processing phase of Storm-Cell Processing begins. Here, decisions are made within each radial as to which groups of range cells are potential candidates as parts of storm cells, without regard to the contents of radials at other azimuths. In the Azimuthal Processing phase, results of the first phase are associated on an inter-radial basis and the storm cells are now defined. In the next phase, the area of each cell is computed, meteorologically insignificant cells are eliminated, and the weighted center of each remaining cell is calculated. The plot phase consists of presenting the centers of the twelve most significant cells together with an outline of the cell itself, along with the centers of all previous cells from as distant as two hours earlier.

Throughout all of these processing phases, the data base remains available on the disc until written over by newly collected data so that further processing, for example, tornado signature detection, can be carried out. The following subsections cover the various processing phases in greater detail than has thus far been presented.

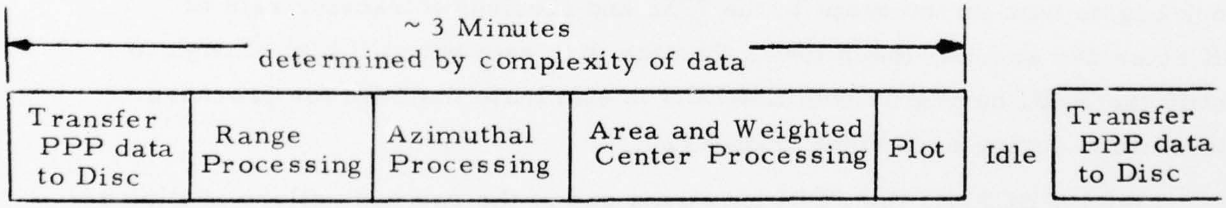

~ 3 Minutes
determined by complexity of data
Transfer PPP data to Disc
Range Processing
Azimuthal Processing
Area and Weighted Center Processing
Plot
Idle
Transfer PPP data to Disc

FIGURE 2-2 TSE SYSTEM TIMING

Times of storm cell processing are not necessarily to scale. Idle time is determined by the user.

2.2.1 Data Collection

Radar video signals processed in the PPP enter the PPP encoder, are combined with ancillary data, then enter the PPRI either in real time or after having been recorded and played back from magnetic tape. The PPP itself imposes limitations on the maximum allowable radar Pulse Repetition Frequency according to the manually-selected N_{RC} (number of range cells) and T_p (range cell width) as listed in Table 2-1. The PPP Encoder requires that the PPP spectral-sweeps-processed control* be set such that $N_{SI} \cdot T_p \geq 64$ µseconds so that adequate time T_d is available to output the serial data. T_d, of course, varies with N_{RC} as shown in Table 2-1. These two limitations have been combined into T_r which represents the minimum time between blocks containing data corresponding to one radial (called frames in the PPP Recorder Manual). In the data collection phase, the PPRI detects a first frame sync, hence generates an interrupt to begin the data transfer for each radial no more frequently than every T_r milliseconds.

An antenna scanning at a uniform angular rate in azimuth should rotate one beamwidth every T_r milliseconds. The optimum scan period for a one-degree-beamwidth antenna is listed in Table 2-1, for the case of minimum T_r (minimum N_{SI}/maximum PRF). This scan period defines the duration of the data collection phase.

*N_{SI} = 1, 2, 4, 8, . . . 512, or 1024.

Table 2-1. PPP/TSE Limitations

No. Range Cells (N_{RC}) →		256	512	768	1024	Limited by
Max. PRF: $1/T_p(N_{RC} + 10)$	①	7518	3831	2570	1934 Hz	PPP
	②	3759	1915	1285	967	
	③	1879	958	643	484	
T_d		15	30	45	60 msec	PPP Encoder
Min. T_r: Min. N_{SI}/Max. PRF		17	33.4	49.8	66.2 msec	PPP and PPP Encoder
Ant. Scan Period @ Max. PRF: 360°×Min. N_{SI}/Max. PRF @ 1° Beamwidth		6.1	12	17.9	23.8 sec	
Bytes/Radial		1040	2064	3088	4112	
Disc Tracks/448 Radials		75	150	224	299	
M Bytes/448 Radials		.466	.925	1.383	1.842	
PPI's/5M Byte Disc		10	5	3	2	

	T_p	Min. N_{SI}
①	.5μs	128
②	1 μs	64
③	2 μs	32

The PPP data in each radial are preceded by 16 bytes of ancillary data, most of which are of interest only at the beginning of each data collection phase. At this time, the time code, N_{RC}, T_p, PRF, and antenna elevation angles are stored in core. The antenna azimuth angle, however, needs to be sampled and stored for every radial. Although 360 azimuth angles should be sufficient for a 1^o beamwidth antenna, a table of 448 halfwords is set up to allow for slight mismatches between PRF, N_{SI} and the antenna scan period. As the antenna scans faster with other parameters held constant, less of the table is used. Each entry in this azimuth table corresponds to one of up to 448 radials of PPP data stored on the disc.

The number of bytes of PPP data per radial varies directly with N_{RC} as shown in Table 2-1. The disc is formatted with 24 256-byte sectors per track, thus between 6 and 1.5 radials fit on each track. A full PPI data set can contain up to 448 radials so that between 2 and 10 sets could fit on the 5M byte removable disc pack, depending on the choice of N_{RC}. More data sets could be accommodated by employing range integration either in software or in hardware.

Data collection may be done in either of two modes: sector scan or PPI. In both cases, no collection takes place until the elevation angle of the radar is within acceptable limits. In the PPI mode, the TSE will begin collecting at the first azimuth at which the elevation angle is acceptable, and terminates when that azimuth is passed again. In the sector scan mode, the TSE will begin collecting data when the azimuth falls between begin and end azimuth limits and the elevation criterion is met.

2.2.2 Storm Cell Processing

Processing of storm data is done in stages: 1) Range processing, 2) Azimuthal processing 3) Area and Center processing, and 4) plot ing of results.

2.2.2.1 Range Processing

During the radar data collection scan, the beginning and end points of meaningful data segments are stored. A segment is considered meaningful if its power z is above the selectable threshold z_{th} for more than ρ range cells in radial extent. These beginning and end points are determined and are stored away for future azimuthal processing. The inner summatic ns:

$$\sum_{z > z_{th}} r \Delta r \tag{2-1}$$

$$\sum_{z > z_{th}} z(r, \theta)\, r \Delta r \tag{2-2}$$

and
$$\sum_{z > z_{th}} z(r, \theta)\, r^2 \Delta r \tag{2-3}$$

are performed at this time also and stored either in the memory or on the disc along with the corresponding beginning and end points. The range element $\Delta r = T_d \cos \phi \cdot 150$ m/μsec represents the flat-Earth ground range of ne radar range cell where ϕ is the antenna elevation angle.

2.2.2.2 Azimuthal Processing

When the data collection scan and range processing are complete l, the beginning and end information is processed to check the azimuthal extent of the area elements. A criterion, α, is established for the minimum number of azimuthally adjacent segments to be considered an acceptable area element. This value, α, is a function of radius ($\alpha \approx \frac{\rho}{r}$) to keep the azimuthal and range criteria approximately equal. As the program steps in azimuth, the extent of segments is cumulated. Once the value α is attained, the set of segments is considered accepted and is assigned a storm cell number. If a given area element fails to reach α, it is unacceptable and is eliminated.

2.2.2.3 Area and Center Processing

When the azimuthal loop is complete, the z-weighted area (analogous to mass if z were mass/unit area)

$$M = \Sigma_A \Sigma z\,(r,\theta)\; r\,\Delta r\,\Delta\theta \qquad (2\text{-}4)$$

is calculated for each storm cell and only the twelve cells with the greatest weighted areas are saved for further processing. The center $(\overline{x}, \overline{y})$

$$\overline{x} = \frac{1}{M}\,\Sigma_A \Sigma\, r^2 \sin\theta\; z\,(r,\theta)\,\Delta r\,\Delta\theta \qquad (2\text{-}5)$$

$$\overline{y} = \frac{1}{M}\,\Sigma_A \Sigma\, r^2 \cos\theta\; z\,(r,\theta)\,\Delta r\,\Delta\theta \qquad (2\text{-}6)$$

for each remaining cell is then calculated.

2.2.2.4 Plotting of results and Track History Storage

If any of the four SCRM display channels are in use for direct display of PPP data, the scaling and origin location are available to the TSE through the SCRMI. The TSE can thus present its output in another channel with the same scaling and origin location as the real time data. Range markers appear as red concentric circles centered about the radar origin location.

At the conclusion of each storm cell processing phase, a unique symbol indicating the coordinates $(\overline{x}, \overline{y})$ of each of the up-to-twelve storm cells is plotted. The actual outline of each storm cell, obtained by converting the beginning and ending points of each segment into the cartesian coordinate system of the display, is also traced-out on the display. The centers obtained in previous processing phases, for up to the past two hours, are also plotted on the display but are color-coded according to time; outlines of past cells are not plotted since they would excessively clutter the display.

More detailed information about past cells is contained in the track history storage area of memory and can be accessed for output on the line printer.

SECTION 3. OPERATION OF TSE

Operation of the Tracking and Significance Estimator may be broken down into three categories. Startup, Execution, and Termination.

3.1 Startup

A complete discussion of the startup procedure for the 7/32 operating system may be found in Reference 1. However, a short summary is included here.

3.1.1 Power Up

To power up the Interdata 7/32 system:

1. Turn on the main circuit breaker in the back of the 7/32 rack.
2. Turn on the Carousel.
3. Turn the power key on the 7/32 processor display panel. If the operating system is still resident in memory, the following message will be printed on the console:

   POWER RESTORE RESET PERIPHERALS AND ENTER GO

   If this message is not printed, refer to Section 3.1.2 to reload the system.
4. Next, turn on the power to the disc drive on the 7/32 rack.
5. When the "LOAD" indicator on the disc drive comes on, mount the "DAVE" volume on the drive.
6. Set the RUN/LOAD switch to RUN and wait for the "READY" light to come on.
7. Enter GO.
8. Turn on Printer
9. Turn on Card Reader (if required)

The system is now ready to run.

Reference 1. Pack/32 Packaging Information Document, Interdata Publication Number 04-072M95R05.

3.1.2 Re-Loading the OS/32-MT System

By following these basic steps, the operating system can be loaded into core from the disc.

1. Load the disc as discussed in steps 4 through 6 in Section 3.1.1.
2. When the "READY" light comes on, enter the following commands, using the hexadecimal display on the 7/32 processor:

```
DTA 7A    ADR
DTA C732  WRT
DTA B6F0  WRT
DTA 930   WRT
```

3. Next, set the "LOAD" switch above the hexadecimal display on.
4. Press INIT.
5. The console should print OS 32MT02-03.
6. After the console has printed, turn the LOAD switch off.

The system is now loaded into core and is ready to run.

3.1.3 Initialization To Run TSE

Before running TSE, the system must be initialized. A complete discussion of the commands used in this section may be found in Chapter 5 of reference 2.

To start the system clock, type

```
SET  TIME  MM/DD/YY,HH:MM:SS
```

where MM/DD/YY is the current date, and HH:MM:SS is the time. This command will ensure that TOCHON and TOCHOFF routines in the I/O drivers work properly.

To mark the discs on, type:

```
MARK  DSC#:,  ON
```

Where DSC#: is either DSC1: or DSC2:. (DSC2: is the fixed while DSC1: is the removeable pack.)

The system should respond with

```
DSC#      VOLN
```

where DSC# is the disc number specified in the MARK command, and VOLN is the four character volume name of the disc, both discs must be marked on to run TSE.

Reference 2. OS/32-MT Program Reference Manual, Interdata Publication Number 29-390R05.

If the system gives an error when marking a disc on,

```
NOFF-ERR  POS  = (DSC#)
```

then the disc must be checked out by the disc integrity check utility (DISCHECK).

3.1.4 Running DISCHECK

A complete discussion of the operation of DISCHECK may be found in Reference 3. A short summary of the command sequence is given here.

To run DISCHECK, enter the following commands:

```
MARK DSC2:,  ON,  P
LOAD  . BG,  DISCHECK
MARK DSC2:,  OFF
TASK   . BG
START       ,  DSC#:,  CON:
```

Where DSC#: is the disc on which the program is to be run.

The console will respond with

```
. BG:  DISCCHECK  :05-00
```

and will run for approximately three minutes.

If there are no errors, the message

```
. BG:  END OF TASK  0
```

will then be printed, and the disc may then be marked on as explained in Section 3.1.3.

3.2 Execution of Tracking and Significance Estimator

In order to run TSE, there must be a source of input, either from the Pulse Pair Processor, or the PPP recorder. To run TSE from the Recorder, the Pulse Pair Processor Interface should be set to the Decoder position, and the Encoder/Decoder Input switch should be set to TAPE. If real-time data from the Pulse Pair Processor is to be used for TSE, the PPP Interface should be set to 'PPP', and the Encoder/Decoder Input switch to REALTIME. For further operating instructions for the Pulse Pair Processor or the PPP recorder, refer to References 4 and 5.

Once an input source has been selected, TSE may be initiated by using the commands discussed in the following sections.

Reference 3. Disc Integrity Check Utility Manual, Interdata Publication Number B29-507R01.

Reference 4. Pulse Pair Processor, Equipment Information Report, Contract F19628-72-C-0293, May 1974.

Reference 5. Pulse Pair Processor Recorder, Equipment Information Report, Contract F19628-74-C-0079, August 1974.

3.2.1 TSE Command

To run TSE, enter the TSE command:

```
TSE [DISP][,BGNA][,ENDA][,ELEV1][,ELEV2][,MODE]
          [,PARFILE][,HIST][,OLDATA]
```

None of the parameters in the TSE command are required. The possible values as well as the default for each operand are discussed below:

DISP = The display to which the output is to be sent. Possible values are 1, 2, 3, or 4. The default is 4.

BGNA = The beginning angle (in degrees) for a sector scan. If a full scan is to be used, this parameter should be null or greater than 360. Possible values are integers from 0 to 360. The default is 400 (full scan).

ENDA = The end angle (in degrees) for a sector scan. Possible values are integers from 0 to 360. Default is 400 (full scan). A more complete discussion of BGNA and ENDA is given is Section 3.2.1.1.

ELEV1 = The lowest acceptable elevation angle (in degrees). TSE will not begin inputting data until the elevation angle is greater than ELEV1 and less than ELEV2. The possible values are integers from 0 to 60. The default is 0.

ELEV2 = The highest acceptable elevation angle (in degrees). Possible values are integers from 0 to 60. However, ELEV2 must be greater than ELEV1. The default is 1 degree.

MODE = The mode in which TSE is to operate. The possible values are:

T = Time option. TSE will input every n minutes, as determined by the parameter file in compliance with BGNA, ENDA, ELEV1, and ELEV2.

D = Demand option - after each run of TSE, the system will pause and wait for a command. When the TCO command is entered, TSE will begin to input data in compliance with BGNA, ENDA, ELEV1, and ELEV2.

C = Continuous operation - Immediately after completing the plot to the scan converter, TSE attempts to input new data in compliance with BGNA, ENDA, ELEV1 and ELEV2. The default is C (continuous operation).

PARFILE = The name of the parameter file. The default is DAVE: TSEDEF. PAR. The format of the parameter file is discussed in Section 3.2.1.2.

HIST = If this parameter is present, the previous track history file (DAVE:TSEHIST.DAT) is used. If this parameter is not present, the file is deleted and a new track history is created. This allows the user to continue track histories between runs.

OLDATA = If the input data was recorded before September 1977, this parameter should be present. If it is present, it tells TSE to ignore the ninth bit of the power data, which was previously used for parity. This allows TSE to be run on both old and new data.

For example, the command

```
TSE 1,,,0,2,,CON:,,OLD
```

will invoke TSE in the following way: the output will go to display 1, a full PPI will be analyzed (default angles are 400, which is greater than 360), at elevations ranging from 0 to 2 degrees. When each run is complete, TSE will immediately begin to input new data (MODE is defaulted to C). The threshold parameters will be input from the console, and the system will prompt the user for each one. The old track history will be deleted and a new one generated, and the analysis will be performed on old data (ninth bit of power is parity).

In another example,

```
TSE ,175,270,,,D
```

will send the output to display 4 (default), for a sector scan from 175 to 270 degrees at elevations between 0 and 1 degree (default). After each run, TSE will pause and wait for the user to restart (DEMAND option). The threshold parameters will be input from the default parameter file, DAVE:TSEDEF. PAR. The old track history will be deleted, and the ninth power bit will be included in the analysis.

3.2.1.1 Sector Scans

TSE automatically determines whether to accept scans in the clockwise or counter-clockwise direction by the sector scan parameters, BGNA and ENDA, of the TSE command. It is assumed that a sector scan is less than 180 degrees.

TSE determines direction by first subtracting BGNA from ENDA. If the result of this subtraction is greater than 180°, it subtracts 360. If the result of ENDA-BGNA is less than -180°, it adds 360. The sign of this result now determines direction, where positive numbers denote clockwise direction, and negative numbers denote counter-clockwise scans.

Thus to scan clockwise from 350° to 50°, BGNA=350 and ENDA=50. To accept data in the reverse direction, BGNA=50, and ENDA=350. To ensure proper operation, the sector scan inputs should be at least 3 degrees inside of the radar's actual sector scan limit.

3.2.1.2 Format of the Parameter File (DAVE:TSEDEF.PAR)

The parameter file may be generated using the system text editor OSEDIT, which is discussed in Reference 6. Each value is a three digit integer, with one value per line. Since this file will be used with a formatted FORTRAN READ statement (I3), all numbers should be right justified.

The default parameter file is as follows:

005	Range Threshold in number of range cells ($0 \le x < NRC$)
005	Power Threshold ($0 \le x < 512$)
005	Range overlap between azimuths = ($0 \le x < NRC$)
010	Percentage of Range to be Ignored (To get rid of ground clutter) ($0 \le x < 100$)
000	Velocity Threshold ($0 \le x < 256$)
000	Variance Threshold ($0 \le x < 256$)
005	Time Between Scans (in minutes ($0 \le x \le 999$)(for time option)

If CON: is used as PARFILE, the system will prompt the user for each input individually with a TSE >.

Each value should be entered as a 3 digit integer followed by a carriage return.

Reference 6. OS EDIT User's Manual, Interdata Publication Number 29-373R03

3.2.1.3 Operating the SCRM

In order to output the results of TSE to the Scan Converter/Refresh Memory, it must be set up properly. Since TSE must read the scaling and origin locations from the SCRM as well as write the plot, both the READ and WRITE Display Memory Access switches on the Display Interface Control panel must be ON. In addition, the MODE switch on the Data Transmission Control panel should be OFF. The STORE VIDEO switch for the display to which the output is to be sent must also be OFF.

A complete discussion of operating procedures for the SCRM may be found in Reference 7.

3.2.2 TCO Command

If the demand option is implemented, TSE will pause after each execution. The console will print:

```
TSE: PAUSE 0
TSE: TASK PAUSED.
```

When the user wishes to continue the task, type

```
TCO
```

followed by a carriage return. TSE will transfer control to PPRDSC subroutine and begin to input data according to BGNA, ENDA, ELEV1, and ELEV2.

3.2.3 TPA Command

If for any reason the user wishes to pause TSE while it is running, type

```
TPA
```

followed by a carriage return. The system will respond with

```
TSE: TASK PAUSED.
```

To continue execution, use the TCO command. The task may also be cancelled by using the TSECAN command, to be discussed later.

3.2.4 DUMP Command

At any time while TSE is running or paused, a track history may be output using the DUMP command. Type

```
DUMP
```

followed by a carriage return. TSE need not be paused to run this program.

Reference 7. Scan Converter and Refresh Memory with Remote Terminal and Display Data Interface, Equipment Information Report, Contract F19628-76-C-0101, August 1976

3.3 Terminating TSE

To terminate TSE after it is paused, type

```
TSECAN
```

followed by a carriage return. TSE must be in the paused state before TSECAN may be used. This can be accomplished by the TPA command. The system will respond with

```
TSE: END OF TASK 255
***TSE  COMPLETE
```

TSECAN will cancel both the TSE and DUMP tasks and re-partition the system to allow other tasks to run. The TCO and DUMP commands are no longer valid once TSECAN is entered.

3.4 Powering Down the 7/32

When the 7/32 computer must be powered down, reverse the order for power up:

1. Mark the discs off:

   ```
   MARK DSC1:, OFF
   MARK DSC2:, OFF
   ```

2. Display Devices to be sure the discs have been marked off:

   ```
   D D
   ```

   The system should respond with:

   ```
   DSC1  C6  OFF
   DSC2  C7  OFF
   ```

3. Set the RUN/LOAD switch to LOAD on the disc drive.
4. Wait about 30 seconds until the LOAD indicator comes on, then turn off the power to the drive. CAUTION: DO NOT TURN OFF POWER UNTIL THE LOAD INDICATOR COMES ON.
5. Turn the power key on the processor to OFF.
6. Turn off the printer and carousel.
7. Turn off the main circuit breaker in back of the 7/32 rack.

SECTION 4. DETAILED HARDWARE DESCRIPTION

This section describes the hardware used by TSE. The new equipment, discussed in Section 4.1, consists of special interface hardware which allows the 7/32 processor to communicate with the PPP encoder/decoder, and the scan converter refresh memory (SCRM). There were also minor changes made to the PPP decoder and the SCRM, which are discussed in Section 4.2.

4.1 New Equipment

Development of the TSE required the design of two special interface devices. The Pulse Pair Recorder Interface accepts radar and Pulse Pair Processor data for further processing in the TSE, while the Scan Converter/ Refresh Memory Interface allows TSE results to be plotted in color on one of the four display channels of the Scan Converter/Refresh Memory.

4.1.1 Pulse Pair Recorder Interface

The Pulse Pair Processor Recorder Interface (PPRI) consists of circuitry added to a Universal Logic Interface Module (ULIM) which plugs into the private bus of an Extended Selector Channel (ESELCH) with device code X'F1' in the Interdata 7/32 processor chassis. The PPRI has only two inputs: a 525K bit/sec serial data line and its corresponding bit clock. The serial data, originating either at the PPP Encoder output or played-back at 30 ips from one track of the FR 2000A wideband instrumentation recorder, includes all radar information needed by the TSE. The data format is described in Figure 2-2 of Reference 5 and consists of from one to four subframes, each consisting of a unique 30-bit synchronization code followed by four 30-bit ancillary data words, followed in turn by 256 30-bit Radar Video data words in a range-cell ordered sequence. The PPRI performs the functions of serial-to-parallel conversion (30 bits at a time), sync code recognition, complementing of certain bits, code conversion, re-ordering of data bits into an Interdata-compatible 32-bit word, and synchronization to accomplish transfer of these words into the processor by means of a DMA transfer through the 16-bit wide ESELCH. The PPRI includes counters to keep track of bits, words, and subframes and also has a detector to permit verification of the presence of the 525 kHz input clock through examination of status bits. A multiplexer on both clock and serial data at the input permits software-controlled testing of this otherwise unidirectional interface.

4.1.1.1 PPRI Data Formats

In transferring 30-bit uniquely-formatted words from the PPP into the TSE which requires 32-bit words or 16-bit half-words, we have decided to perform data conversion and bit re-ordering in the PPRI hardware in order to minimize software bit-manipulation requirements. In this process, video and ancillary data must be handled differently because of their different codes and formats. Figure 4-1 illustrates the format for video data in which the first ten Encoder output bits B0:9 (also M0:9 in Reference 5) represents Mean Doppler Velocity, B10 (M10) is the sign bit, and B11 (M11) is a parity bit over all Mean bits except B1 (M1). The next eight bits B12:10 (V0:7) represent spectral Variance, an unsigned quantity, while B20 (V8) is the parity bit for B12:19. Bits B21:28 (P0:7) convey logarithmic radar return power, also unsigned, with the last bit B29 (P8) as parity. However, AFGL sometimes uses this ninth power-bit as an additional data bit to extend dynamic range.

Figure 4-1 indicates a mapping of Encoder Video Data output bits into Interdata fullword bits which results in the format shown in Figure 4 -2 for the data in 7/32 core, and if fullword boundaries are observed, in processor registers loaded from core. There may be confusion because Interdata bit numbering is the reverse (the LSB is the highest-numbered bit) of that used in the PPP and Encoder. The notation D0:10 represents Mean Doppler velocity M0:10 converted (in PPRI hardware) to two's complement with the sign bit D10 positioned as required for halfword arithmetic instructions. If the eight most-significant Doppler-velocity bits are adequate, only a Load Halfword instruction followed by masking of bits 24:31 (containing Variance-unwanted in this case) is required. Obviously, if Variance data were wanted, Bits 16:23 could be masked instead. All Doppler velocity bits can be gotten together by execution of these instructions, for example: Load (fullword), Exchange Byte Register, Rotate Logical Left (8 bits), And Immediate (to mask unwanted bits) and Convert to Halfword Value Register. Eight bits of Power or Variance can be obtained simply through use of Load Byte instructions, while nine bits of power can be loaded by executing a Load Halfword then by masking of bits 0:6.

While this scheme seems to provide a format reasonably compatible with subsequent software manipulation of video data, it would hopelessly scramble Ancillary data which contain a mixture of BCD, binary and special coding. For this reason, Ancillary data need a different bit re-ordering as indicated at the bottom of Figures 4-1 and 4-2. It might be nice to reverse Hours and Minutes, but this would add hardware complexity since bits

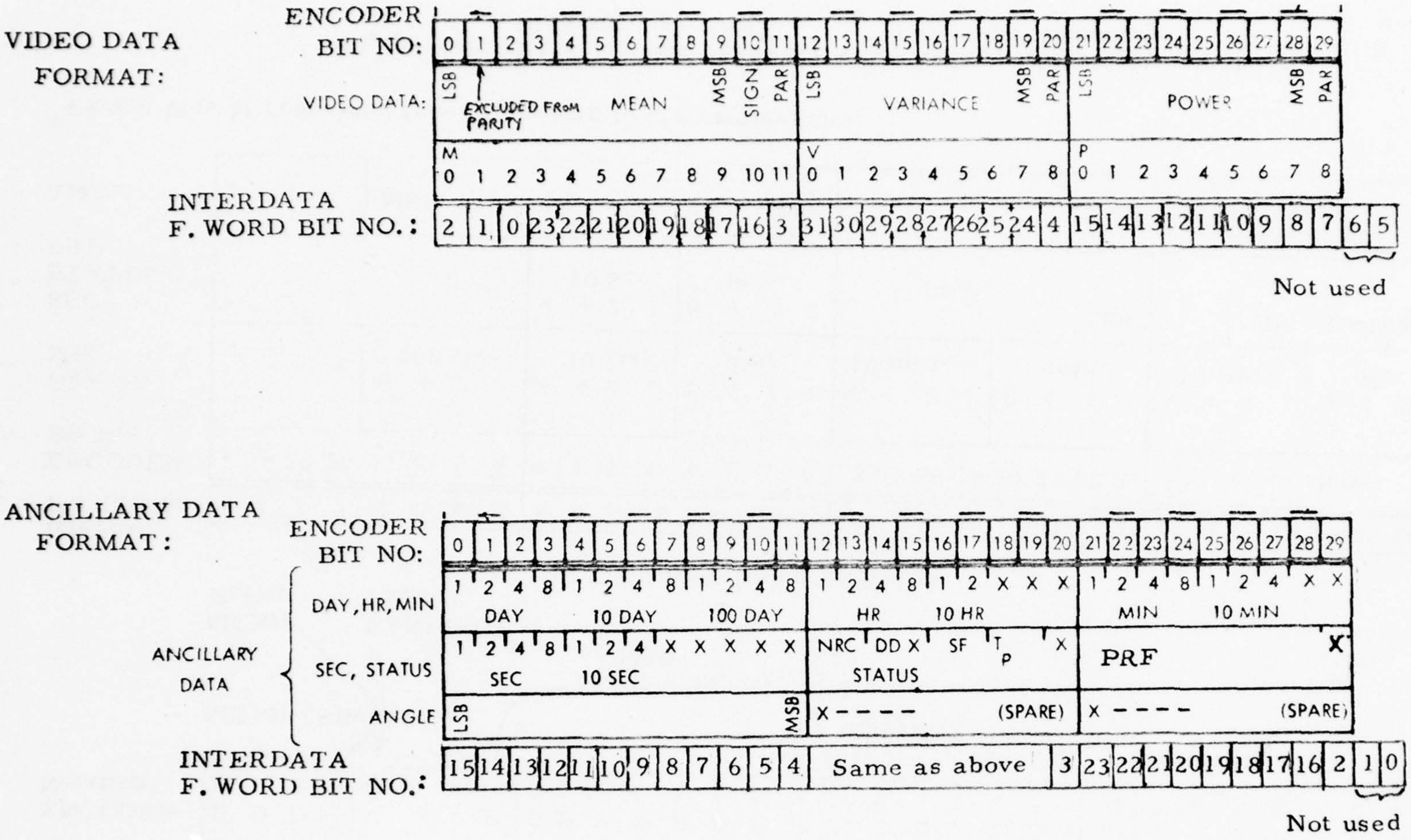


Figure 4-1. Encoder Output Formats Showing Re-Ordering of Bits into Interdata Fullwords

VIDEO DATA FORMAT

																																LSB
Interdata FW Bit No:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ENCODER Bit No:	2	1	0	11	20	-	-	29	28	27	26	25	24	23	22	21	10	9	8	7	6	5	4	3	19	18	17	16	15	14	13	12
ENCODER Notation*	D 2	D 1	D 0	M 11	V 8	- -	- -	P 8	P 7	P 6	P 5	P 4	P 3	P 2	P 1	P 0	D 10	D 9	D 8	D 7	D 6	D 5	D 4	D 3	V 7	V 6	V 5	V 4	V 3	V 2	V 1	V 0

MEAN LSBs

MEAN Parity

VARIANCE Parity

Spare

POWER Parity or MSB

LSB

Sign

MEAN DOPPLER VELOCITY

VARIANCE

LSB

ANCILLARY DATA FORMAT

																																LSB
Interdata FW Bit No:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ENCODER Bit No:	-	-	29	20	11	10	9	8	7	6	5	4	3	2	1	0	28	27	26	25	24	23	22	21	Same as above							
Day, Hr., Min.					8 100 Day	4	2	1	8 10 Day	4	2	1	8 Day	4	2	1	x 10 min.	4	2	1	8 min.	4	2	1	x 10 Hr.	x	2	1	8 Hr.	4	2	1
SEC, STATUS, PRF									x 10 Sec.	4	2	1	8 Sec.	4	2	1	PRF							LSB	T_p		SF		x	DD	NRC	
ANGLE					MSB											LSB																

*Except D is M converted from sign-mag to 2's complement.

Figure 4-2. Interdata full words for video and ancillary data entered via DMA through PPRI from PPP Encoder.

24:31 could not be handled identically between Video and Ancillary formats. This reversal can be done very simply in software by execution of the Exchange Byte Register instruction. The azimuth and elevation angle words, coded in binary with a MSB weight of 180 degrees, appear as the twelve least-significant bits of a halfword.

4.1.1.2 PPRI Hardware Description

The Serial Data and 525 kHz Clock inputs of the PPRI are TTL-level compatible with increased noise immunity achieved through use of Schmitt-trigger (hysteresis) inputs. They are resistively terminated with 100-ohms to approximately match coaxial lines from the PPP Recorder Encoder/Decoder. The Encoder Data state changes after the high-to-low transition while the PPRI samples the Serial Data on the low-to-high transition of the clock, so that sensitivity to clock skew is minimized.

A block diagram of the PPRI is included as Figure 4-3, where the inputs just discussed can be seen in the upper left corner. A multiplexer provides for software-controlled testing to be discussed later; for the present, assume that the Encoder outputs drive the Serial Data and Clock lines shown connected to a 30-bit SIPOR (Serial-In/Parallel-Out Register). In the Encoder, certain bits are complemented to ensure an adequate number of bit transitions in the case of all-zeroes or all-ones data, so that the phase-locked loop which reconstructs the bit clock on playback in the FR 2000A can operate properly. These bits need to be complemented again at the SIPOR output in order to restore them to the proper logic sense. The fact that both complemented and true versions of these bits are available simplifies the design of the sync decoder which recognizes both the first sync code (30 bits) and any sync code (28 bits). Detection of the first sync code (A) resets the modulo 4 subframe Counter and generates an interrupt through the General Purpose logic on the ULIM. Detection of any sync code (A, B, C or D) resets the Input Bit and Word Counters to their respective proper starting states.

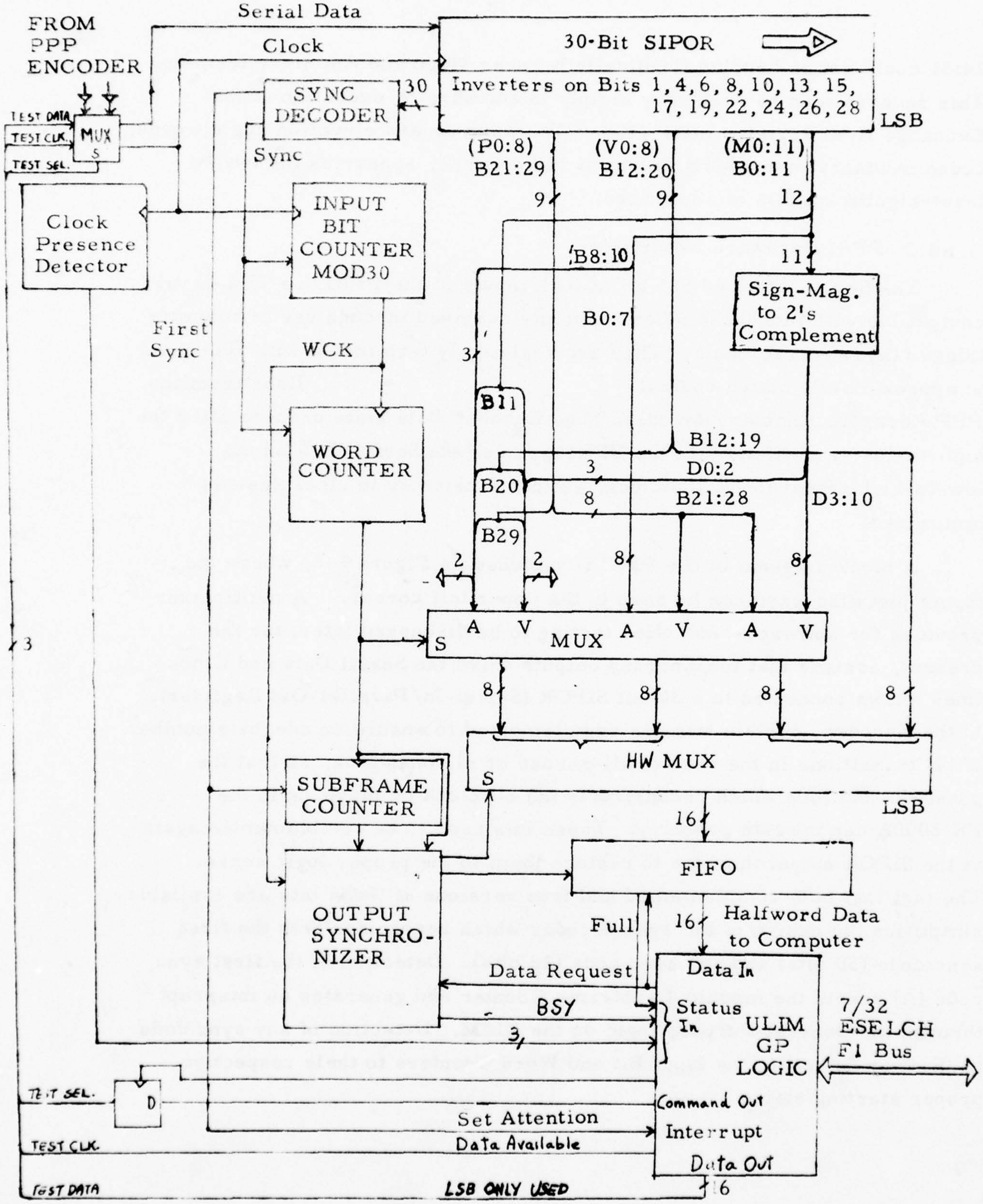


Figure 4-3. PPRI Block Diagram

After the appropriate SIPOR output bits have been complemented, they are routed through an interconnection matrix to a 16-bit two-input "Halfword" multiplexer via a Sign-Magnitude to two's complement converter for the Mean Doppler velocity bits and a 24-bit two-input multiplexer which implements the two different bit-reordering schemes needed for Ancillary and Video data. This multiplexer's select line is driven by a signal decoded from the Word Counter which keeps track of whether the word being received contains Ancillary or Video data. A Subframe Counter prevents the transfer of Ancillary data except during the first subframe by disabling the Output Synchronizer during the Ancillary data words of subframes B, C and D. Just after each 30-bit serial word has been shifted into position in the SIPOR, allowing of course for propagation delay through the code converter and multiplexer, the Output Synchronizer sequentially enters the two halfwords now at the HW Multiplexer inputs into the FIFO (First-In/First-Out) buffer memory.

Because the memory data bus within the 7/32 is 16 bits wide, transfers to or from this memory take place in halfwords via the ULIM GP Logic connected to the 7/32's ESELCH bus. The halfwords in FIFO are already in the proper order for transfer to the ESELCH, so the output synchronizer must now simply control the transfer rate. The synchronizer accomplishes this control by clocking halfwords out of the FIFO by issuing appropriately-timed transfer Request pulses to one of the status bits which is interpreted as the BSY bit by the ESELCH. The peak transfer rate is 10^6 halfwords/second when two successive halfwords resulting from each 30-bit input word are outputted, but these halfword pairs are 57 μseconds apart (except for a special case following interrupt processing) resulting in an average rate during most of each block 17,500 fullwords or 35,000 halfwords/second so that only 3.5% of the one-microsecond memory cycles are stolen. The overall average transfer rate, at the maximum PRF possible, is about 86 to 88% of the peak rate because of dead time and the gaps left by sync codes and ancillary data from subframes B, C and D. Of course, at lower PRFs, the overall average transfer rate will decrease.

To begin the data transfer, a command byte with bits 1 and 4 set must be issued to the PPRI (device code X'8A') to clear the clock presence detector. Once the detection of a first sync pulse results in the generation of an interrupt, the 7/32 has up to 0.5 milliseconds in which to set up the ESELCH (device code X'F1') while halfwords accomulate in the FIFO. The ESELCH should be set up with the proper starting and ending addresses and given a GO command. When the ESELCH begins to run, it finds BSY low whereupon it generates a Data Request which clocks a halfword out of the FIFO and into the ESELCH. The ESELCH sends another Data Request pulse as fast as it can steal another one-microsecond memory cycle. This process continues until all halfwords which had piled up in the FIFO during interrupt processing time have been clocked into the ESELCH; now the Output Synchronizer sets BSY high. As each subsequent halfword enters the FIFO, BSY goes low, the halfword is transferred into the ESELCH, then BSY goes high again. Thereafter, until the transfer process is terminated, two halfwords pass through the FIFO every 57 μseconds and never pile up.

A Clock Presence Detector, implemented using retriggerable monostable delay elements, verifies the presence and approximately-correct frequency (525 kHz) of the input clock. The outputs of the Presence Detector control ESELCH termination and can be examined by the processor as two of the eight status bits through the ULIM GP Logic. Status and Command bytes for the PPRI are summarized in Table 4-1. As has been discussed, status bit 4

Table 4-1. PPRI Status and Command Bytes

	0	1	2	3	4	5	6	7
Command	DIS	EN		TEST	CLR			
Status					BSY	FIFO FULL	CLK FAST	CLK SLOW

Device Code X'8A'

Normal	0	0
Clock slow or stuck High	0	1
Clock fast	1	0
Clock stuck low	1	1

controls the ESELCH transfer rate. If any of bits 5, 6 or 7 become set during a transfer because of some abnormal condition, then that transfer is terminated.

Command bits 0 and 1 control the interrupt queuing circuits in the ULIM GP Logic, while bit 3 selects the TEST position of an input multiplexer. Command bit 4 clears the Clock Presence detector: initially, both Status bits 6 and 7 go high. Within several μseconds, these bits indicate the condition of the clock as listed in Table 4-1. Bit 4 resets the PPRI sequential logic.

The previously mentioned multiplexer at the Serial Data and Clock inputs permits software-controlled testing of the PPRI by the following procedure. A command byte having a bit assigned to control the input multiplexer is issued to the PPRI via the Multiplexer bus -- the ESELCH is not used in this test. This Test Select bit, stored in a D-Flop shown at the lower left of Figure 4-3, applies the Data Available strobe from the GP Logic and the LSB only of the Data Output to the SIPOR clock and data inputs, respectively. Now the processor can generate sync codes and simulated data in software and can verify proper serial/parallel conversion, bit complementation, sync decoding, code conversion, bit reordering, counting, and output synchronization by observing outputs from the halfword multiplexer. See Appendix A.

Not shown in Figure 4-3 is the System Clear GP Logic output which resets all sequential logic elements in the PPRI to drive it to a known starting state.

4.1.1.3 Data Transfer

An example of a data transfer from the PPP recorder to the computer is shown in the timing diagrams of Figures 4-4 and 4-5.

Before all data transfers, the programmer outputs a clear and interrupt enable commands to the PPRI. The clear command clears the FIF0, and prevents data from entering the FIF0 until a first sync code is recognized (SYNC1). CLR also initializes the clock presence detector, which is cleared to 00000011 (status bit 6 and 7 high). Within 2 to 4 μ seconds, the clock presence detector should have

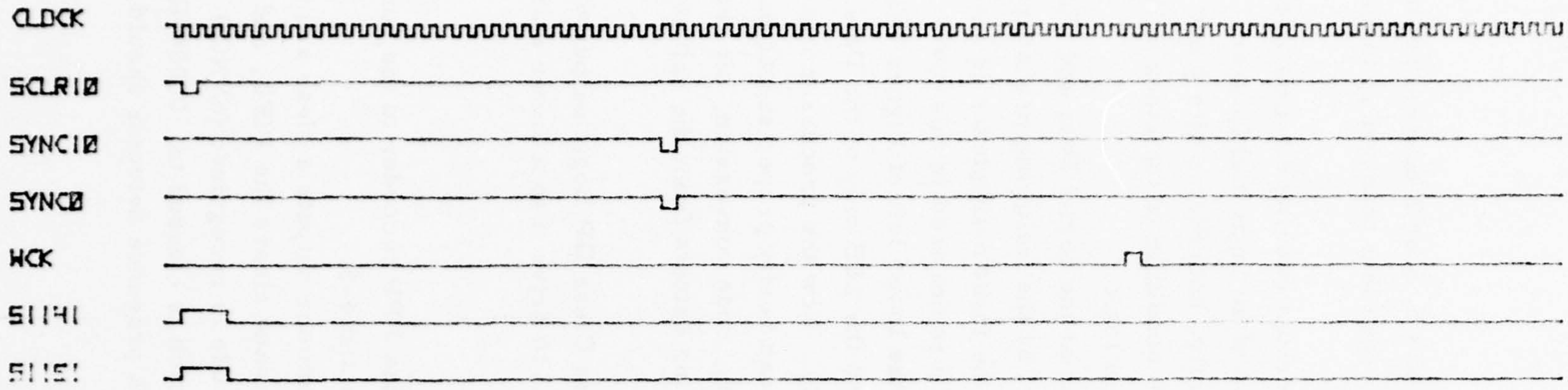


FIGURE 4-4.TIMING DIAGRAGM FOR INITIALIZATION AND STARTUP OF PPRI.

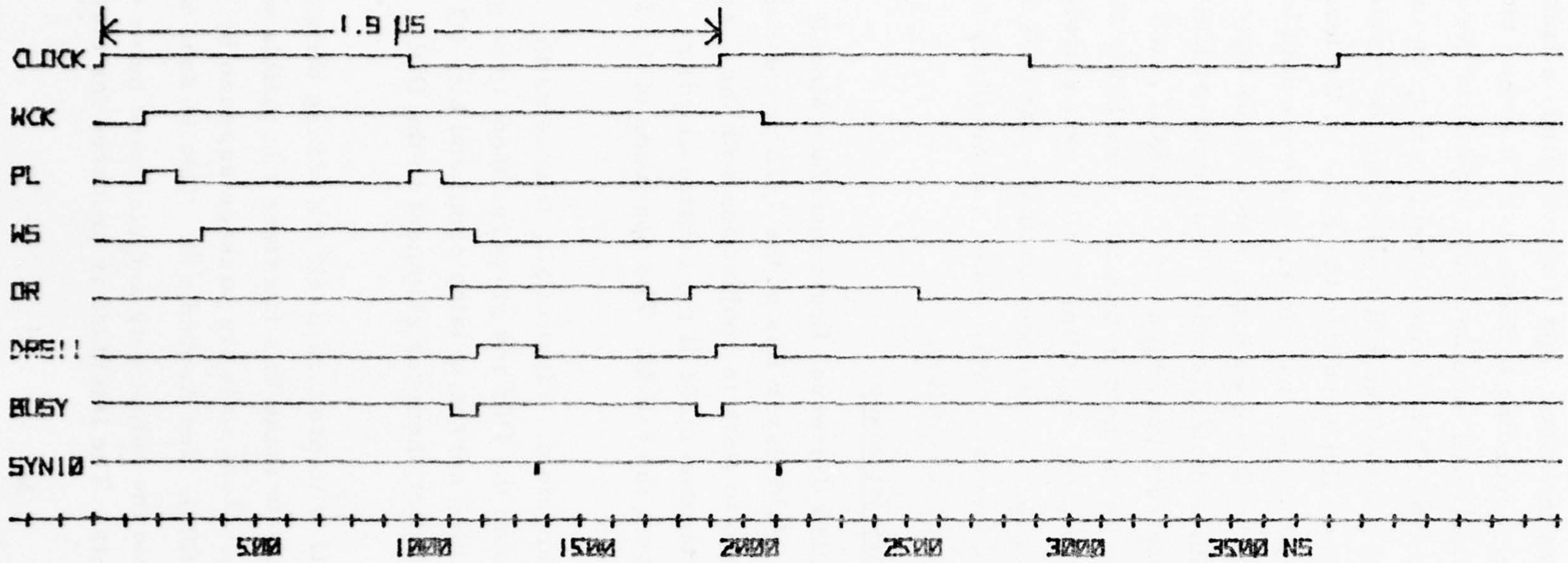


FIGURE 4-5. TIMING DIAGRAGM SHOWING FIFO
TIMING IN APRI.(ACTUAL DELAY FROM PL TO
DR HIGH IS 8 US)

valid data. If either status bit 6 or 7 is high, or both, then data transfer cannot occur. The CPD continually monitors the clock, and if it ever finds a fault in the clock input, bit 6 or 7 will go high according to Table 4-1. If either bit goes high during a data transfer, the ESELCH will automatically terminate the operation and output an error status. Once the SYNC1 code is recognized, an interrupt pulse is sent to the processor, signalling that the ESELCH should be started. Load pulses (PL) are generated to load each word into the FIFO on the leading edge of WCK and the next trailing edge of the input clock. Once the data has rippled through the FIFO, Output Ready (OR) goes high, which sends busy low. The ESELCH senses BUSY low and immediately sends a Data Request (DRG11) pulse to the PPRI, which subsequently sets busy high again. DRG11 gates the data onto the bus, while low, and clocks the next 16 bits out of the FIFO when it goes high again. When OR goes high again, signalling that the next halfword is in position, BUSY goes low, and the process continues until the ESELCH determines that all the data has been transferred. Sync codes and ancillary data in all subframes except the first are not transmitted.

4.1.2 Scan Converter Refresh Memory Interface

The SCRMI consists of a modified Universal Logic Interface Module (ULIM) which plugs directly into the Multiplexer Bus of the 7/32 processor, with device code X'8C'. It is designed to work in conjunction with the Display Data Interface DDI) which already contains most of the control circuitry necessary for data transfer to and from the SCRM. The operation of the DDI is discussed in section 4.2.2.

The function of the SCRMI is threefold. First of all, it accepts the status byte and 8 bits of data input from the DDI and prepares them to be gated onto the 7/32's Multiplexer Bus. It also serves to latch command bits 4:7 and the data byte from the 7/32 and buffer them for shipment to the DDI, along with the control signals.

The third function of the SCRMI is to prepare itself for testing through software. It was considered desirable to make this interface compatible with the Common Universal Logic Interface Test software package supplied by Interdata (program number B06-129R06). See Appendix B. This is done with the use of multiplexers which logically tie the output byte and command bytes 4:7 of the SCRMI with the status and data inputs. The interface is prepared for testing by setting command bit 3 to a 1.

The SCRMI is connected to the DDI through 33-foot long ribbon cables. The cables are wired in an alternate ground configuration to minimize cross talk.

4.2 Existing Equipment

The TSE has been designed to minimize changes required to be performed on existing equipment.

4.2.1 PPP Recorder

The serial data and clock signals needed to drive the PPRI, selectable by the DATA SOURCE switch between REAL TIME (Encoder) and TAPE switches, are both available on the Decoder card. A line driver, Type SN 74128, is added there along with output connections to two BNC connectors on the rear panel. From these points, two RG62 93-ohm coaxial cables route the serial data and clock signals to the PPRI.

4.2.2 Scan Converter Refresh Memory

The SCRMI in the TSE communicates with a Display Data Interface in the SCRM always in the byte mode. No DDI modifications are needed to perform the data transfers required by the TSE, but this section briefly outlines the DDI's capabilities and references other documents for details.

The 7/32 treats the Scan Converter as a peripheral and can perform data transfers to and from its refresh memories through the ULIM with device code X'8C', the SCRMI on the ULIM, and the DDI within the SCRM. In addition to these processor-initiated operations, a cursor can be switched on in any display, its position adjusted by means of a trackball, and the four-bit color code in the refresh memory at that position transmitted to the computer by operator interaction through the Display Interface Control panel.

Manual control of the DDI, through the Display Interface Control panel and various controls on the Scan Conversion Processor itself, is instructed in section 3.1.10 of Reference 7. Section 4.7 of that document presents a detailed hardware description of the DDI and how it interacts with other components of the SCRM. The three basic types of transfers: Write Display Memory, Read Display Memory and Cursor Data Entry are exemplified in section 4.7.7 which presents hardware-oriented discussions of the same examples described in software terms in Appendix C of Reference 5.

The DDI Command and Status Bytes are listed in Table 4-2. Command bits 0 and 1 program the interrupt logic on the ULIM to either accept, queue, or ignore interrupts from the DDI. The HW bit is used to select halfword or byte mode in Interdata ULI cards, but since this SCRMI is built on an MDB System ULIM, this bit is neither needed nor used because the DDI always operates in the byte mode. Command bit 3 enables the test mode described in section 3.2

Command bits 4, 5 and 6 direct control logic on the DDI to perform various data transfer sequences. In the WRITE operation, the 7/32 specifies the display channel, the color of the picture element to be written, and the address. Three different sequences are available, depending on whether successive elements are to be the same or different colors.

In a READ operation, the 7/32 specifies the display channel and the address and receives from the DDI the color code for the addressed picture element. Since all WRITE operations require random access to the refresh memory of the selected channel, no WRITE sequence should be attempted if the Not-Available status bit for that channel is true (Table 4-2). If, however a READ operation is initiated in a channel with its Memory Bus not available ($\overline{MBA}$ = true), the operation will still be accomplished by a SLOW READ process in which the required picture element is obtained by waiting for it to appear in the digital video data stream used to refresh the raster-scan display. An interrupt is generated at this time. SLOW READ can also be purposely initiated by issuing the appropriate command byte.

Table 4-2. DDI Command and Status Bytes

	0	1	2	3	4	5	6	7
Command Byte	DIS	EN	HW	TEST	COT4	COT5	COT6	COT7
	To Interrupt Control Logic on ULIM Device Code X'8C'		Not Used		To DDI Control Logic			Spare to DDI

COT-

4	5	6	7	
0	0	0	X	WRITE Multiple Points Same Color
0	0	1	X	" " " Different Colors - Not fullword Boundaries
0	1	0	X	" " " " " - On " "
0	1	1	X	
1	0	0	X	READ (SLOW READ FORCED IF $\overline{MBA}$)
1	0	1	X	SLOW READ
1	1	0	X	
1	1	1	X	INITIALIZE CONTROL LOGIC

	0	1	2	3	4	5	6	7
Status Byte	1NA	2NA	3NA	4NA	SDG	WS	RS	
	Display Channel Not Available				Send Data Gate-A Cursor Data Entry has been initiated.	WRITE Switch ON	READ Switch ON	

In a manually-initiated Cursor Data Entry operation, the color code is obtained as in SLOW READ but the address comes from the trackball and is transferred to the 7/32 along with the color and channel codes when the interrupt happens. When a cursor data entry is in process, status bit 4 is true, even before the interrupt has occurred.

Switches on the Display Interface Control panel can disable either READ or WRITE operations in the DDI. The state of these switches is available to the 7/32 through the status byte.

Details of the various types of transfers, includi ng hardware and software examples, display addressing conventions, and control logic operation ar e available in section 4.7 and Appendix C of reference 7 and should be stud ed before programming is attempted.

SECTION 5. DETAILED SOFTWARE DESCRIPTION

5.1 Support Software

The support software used by TSE may be broken down into two categories: the OS/32-MT operating system, and the I/O drivers. These combine to form a powerful aid to both the programmer and the end user.

5.1.1 Operating System

The choice of the operating system was dependent upon the language to be used for the task to be performed. Since FORTRAN was chosen for the task, this mandates that an Interdata operating system also be used. Appendix F, an Interdata product bulletin entitled OS/32 MT, describes in detail the operating system which was used for TSE. In addition, extensive use was made of the Command Substitution System. The Command Substitution System (CSS) is an extension to the OS/32 MT Command Language. It provides the user with the ability to establish files of dynamically modifiable commands which can be called from the console or other CSS files and executed in a defined sequence. In this way, complex operations can be carried out by the operator with only a small number of commands.

CSS provides more than just the ability to switch the operating system command input stream to a batch device:

- A set of logical operators are provided to control the precise sequence of commands to be obeyed.
- Parameters can be passed to a CSS file so that general sequences can be written which take on specific meaning only when the parameters are substituted.
- One CSS file can call another, in the manner of a subroutine, so that complex command sequences can be developed.

A CSS file is simply a sequential text file. It could be a deck of cards, a punched paper tape, a magnetic tape, or a disc file.

The OS/32-MT High Level Operator Command Package is implemented as a set of CSS files. These perform a variety of the commonly used program preparation and development sequences. The following are some of the more often used files:

Command	Description
FORT	Perform FORTRAN V compile, assembly and task establishment
FORTCLG	Perform FORTRAN V compile, assembly, task establishment, load and start
CAL	Perform CAL assembly and task establishment
CALCLG	Perform CAL assembly, task establishment, load and start
COPY A	Copy an ASCII file using OS Copy
COPYB	Copy a Binary File using OS Copy
COPY T	Copy an established task, resident library or overlay using OS Copy
RUN	Load and start a task.

5.1.2 Drivers

In order to perform useful work, a program may have to communicate with many different peripheral devices, each with its own programming requirements. To provide the programmer with the ability to perform I/O to different devices without requiring device dependent coding, OS/32 provides a standard I/O interface. Figure 5-1 illustrates the elements of this I/O interface. I/O requests are initiated by a Supervisor Call (SVC 1); drivers provide the device dependent support via routines and control blocks, and access to the devices is controlled by the Event Service Handler through the Event Coordination Table (EVT).

Two additional drivers were required for inclusion into the OS/32 MT operating system. These drivers each control one Universal Logic Interface Module (ULIM); one for the Scan Converter Refresh Memory Interface (SCRMI) and the other for the Pulse Pair Processor Recorder Interface (PPRI). Each driver was written to conform to the requirements specified by Interdata for user-written drivers.

5.1.2.1 PPRI

This port provides an input-only interface between the Extended SElector CHannel (ESELCH) bus and the Pulse Pair Recorder. Two modes of data transfers are used: Command and Status bytes via the multiplexer bus, and data via the ESELCH in the half-word mode on the DMA bus.

5.1.2.1.1 Supported Attributes

Read, Wait, I/O and Proceed. Unconditional proceed cannot be supported by this device because the status byte does not provide information as to device availability.

5.1.2.1.2 Functional Description

Upon entry to the driver at the entry point INITPPRI, the interrupt is enabled. Refer to Table 4-1 for the Command and Status bytes. Bit 4 of the Command byte, clear, is also output at this time to initialize the PPRI. The ISP is set to point at the CCB for the PPRI driver. Upon receipt of the interrupt, a command byte of X'C0' is output to disarm the PPRI interrupts. The status byte is examined to determine if one of the error bits in bits 5, 6, or 7 is set. If so, an error code as defined in section 5.3.1.3 will be placed in the returned status variable in the DCB for output by system routines to the operator's console and the termination routine is scheduled. If no error, the first and last byte addresses are loaded into the ESELCH with the address of X'F1' and a GO command is given to the ESELCH to start the data transfer. The address in the ISP for device X'F1' is set to point at the interrupt routine for the PPRI ESELCH. Upon receipt of the ESELCH interrupt, a STOP command is issued to the ESELCH to stop the data transfer if the interrupt resulted by an error bit being set. The termination routine, which is executed in an interruptable state, is scheduled by the ESELCH ISR.

The termination routine, entry point TERMPPRI, inputs the status byte, converts the status to the proper code and stores it in the DCB for retrieval by system error routines or north-seeking routines, and computes the length of the last transfer by reading the last byte address transferred by the ESELCH and subtracting the first byte address from the parameters in the SVC1 parameter block, and an error is generated if the values do not match.

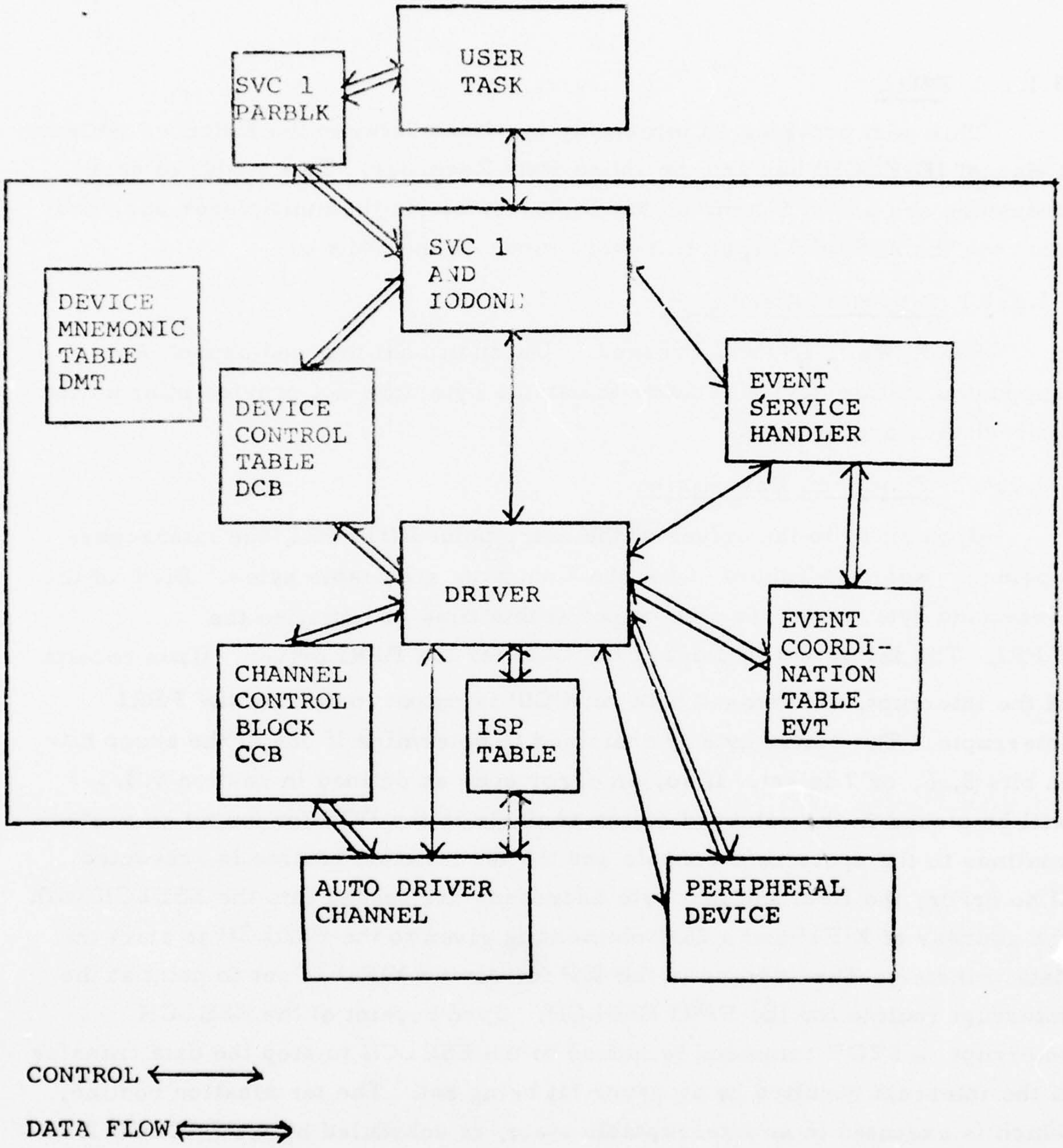


Figure 5-1. Elements of I/O System

Operating system routines such as timer routines TOCHON and TOCHOFF, and completion routine IODONE were used to insure compatibility with the operating system as described in Figure 5-1. The time-out constant is set for 2 seconds after which, if no I/O has begun, an error will be generated.

5.1.2.1.3 Status Definition

Status	Meaning
X'00'	Normal Completion
X'11'	Clock slow or stuck high
X'12'	Clock fast
X'13'	Clock stuck low
X'82'	Time out
X'84'	Short data transfer
X'C0'	Illegal function

5.1.2.1.4 Device Address

The device address of the PPRI is X'8A'.

5.1.2.1.5 Parameter Block

The PPRI driver is called by a supervisor call (SVC1) with a standard Interdata I/O parameter block as discussed in Chapter 4 of Reference 2.

5.1.2.2 SCRMI/DDI

This port provides a transparent interface between the Multiplexer bus and the Display Data Interface (DDI) in the SCRM. Command, status, and display output data bytes are transferred via direct commands while the display input data can be transferred via direct commands and the Auto Driver channel.

5.1.2.2.1 Supported Attributes

Read, Write, Wait, I/O and Proceed. Unconditional proceed cannot be supported by this device because the status byte does not provide information as to device availability.

5.1.2.2.2 Functional Description

Upon entry to the driver at the entry point INITSCRM, the function code, located in DCB.FC, is tested to determine the type of request; READ or WRITE.

5.1.2.2.2.1 READ

Due to the uncertainty of the availability of the DDI Memory Bus, all read operations will be performed in the "SLOW READ" mode. A command byte of X'CE', disarm interrupts and initialize, is output at this time. The ISP is set to point at the CCB for the SCRMI driver and the CCB is initialized with the starting address, length, and subroutine address. A SINT is executed to force the driver into the IS state and start the I/O process.

Upon receipt of the first interrupt, a command byte of X'48', interrupt enabled and slow read, is output. Three WD commands are executed to transfer the X and Y addresses to the DDI and the routine exits to await the data interrupt. Upon receipt of the interrupt at entry point ISR2SCRM, the color is read and stored into the users buffer, the X and Y addresses are incremented by the values stored in the SVC1 parameter block, and the number of points input is incremented. This process continues until the desired number of points has been transferred at which time the termination routine is scheduled.

5.1.2.2.2.2 WRITE

Three modes of write operation are possible with the DDI. The main differences are in the number of write commands required. This allows the programmer flexibility in structuring data buffers for output. All output will be done in the initializer due to the fact that no interrupts are generated by the SCRMI/DDI during the output phase.

Upon entry to the WRITE portion of the driver, the WRITE switch, bit 5 of the status byte, is tested to determine if the DDI cannot accept data; if so, an error status is stored and the program exits. The requested display number is used as an index into the status byte bits 0 through 3 to determine if the desired display channel is not available for output; if so, an error status is stored and the program exits. Once it has been determined that the display is ready for output, the program enters the routine to output the data dependent upon the mode of output desired. Figure 4 of Appendix C of reference 7 depicts the modes and sequence of commands for the various output modes. The output loop will transmit data until the desired number of points have been output.

5.1.2.2.2.3 TERMINATION

Upon completion of the READ and WRITE routines, the termination phase is scheduled to complete the housekeeping and provide an orderly exit to the operating system. Operating system routines such as timer routines TOCHON and TOCHOFF, and completion routine IODONE will be used to ensure compatibility with the operating system as described in Figure 5-1. The time-out constant is set for 2 seconds after which, if no I/O has begun, an error will be generated.

5.1.2.2.3 Status Definition

Status	Meaning
X'00'	Normal Completion
X'82'	Time out
X'83'	Recoverable Error-Write Switch Off
X'An'	Device Unavailable-Write channel n
X'C0'	Illegal Function

5.1.2.2.4 Device Address

The device address of the SCRMI/DDI is X'8C'.

5.1.2.2.5 Parameter Block

The SCRM Driver is invoked by a Supervisor Call (SVC 1) with a standard Interdata parameter block as discussed in Chapter 4 of Reference 2. In addition to this, the buffer contains information used by the driver. The format of the buffer is shown in Figure 5-2.

NUMPTS is a full word specifying the number of points to be written to or read from the display.

X-COORD is a nine bit number specifying the number of columns from the leftmost edge of the display. There are 320 horizontal positions on each display. If the number is greater than 360, it is ignored.

Y-COORD is an eight bit number specifying the number of rows from the uppermost edge. There are 248 vertical positions on a display. If the number is greater than 248, it is ignored.

FC is the function code which specifies what type of I/O is to be performed. If FC is a '1', the driver performs I/O on a horizontal line of length NUMPTS, starting at (X-COORD, Y-COORD). If FC is a '2', I/O is performed on a vertical line of length NUMPTS beginning at (X-COORD, Y-COORD). If FC is a '3', I/O is performed on NUMPTS random points whose individual coordinates are given in the fullwords which follow NUMPTS.

INCR is an increment value which acts on either X-COORD or Y-COORD, depending on the value of FC. For example, if FC = 1, and INCR = 2, and a read operation is performed, every other point on a horizontal line beginning at (X-COORD, Y-COORD), will be read into the buffer. If INCR = 3, every third point will be read. If INCR = 0 or 1, every point will be read. If FC = 3, this option is ignored.

If the most significant bit of the display byte (bit 16 in Figure 5-2) is a 0, and the display is unavailable, the driver will automatically search for the nearest available display. If all four displays are unavailable, an error condition results. If bit 6 is a 1, this option is disabled, and if the display specified is unavailable, an error condition will be returned. In both cases, the value in the display field is left unaltered.

	0 7	8 15	16	17 23	24 31
WORD 0	—	INCR	AUTO	DISPLAY	FC
1	NUMPTS				

	0 3	4 6	7 15	16 23	24 31
2	COLOR	—	X-COORD	Y-COORD	—
3	COLOR	—	X-COORD	Y-COORD	—

•
•
•

FIGURE 5-2 BUFFER FORMAT FOR SCRM DRIVER.

The begin and end addresses of this buffer appear in the standard SVC 1 parameter block.

5.2 Task Description

The TSE system consists of four tasks: TSEPARAM, TSE, TSEDUMP, and CHANGE. TSEPARAM runs as a background task (.BG) before TSE is loaded into memory. TSE runs in a foreground partition (TSE) and requires 87K* of core. Simultaneously with TSE, either of two tasks, TSEDUMP or CHANGE, may run. TSEDUMP is the largest of the two, and requires 5K of core to run. Therefore, a second foreground partition of 5K is set aside for this purpose.

There are also four task common blocks (see Appendix C) used by TSE, to allow intercommunication between tasks. These blocks are set up in global memory under the names TASK, ZSTORE, DONE, and EXTRA. In all, they occupy 16K of memory. Since this memory is global memory, it cannot be executed, thus limiting the executable memory to 176K.

In order to utilize memory most efficiently, a run-time library containing many of the commonly used FORTRAN subprograms was generated. This run-time library is called TSERTL.RTL, and resides on the TSE system disk (DAVE). TSERTL must be present in core before TSE can be loaded, and resides in the .LIB partition. It may be loaded using the LOAD command:

```
LOAD .LIB,DAVE:TSERTL.RTL
```

TSERTL occupies 7.75K of memory. To erase the .LIB partition, and free the memory for other purposes, use the SET PARTITION command:

```
SET PARTITION .LIB/0
```

These commands are discussed in full in Chapter 5 of Reference 1.

The following sections describe each routine in TSE, in the order in which they are executed. They are classified by their file names, which may be slightly different than their routine names.

5.2.1 TSEPARAM.CAL

TSEPARAM is a TSE support program which is called by the TSE command. It is an assembly language program, and operates in approximately 1.25K of memory. Its purpose is to transfer the parameters of the TSE command to TSE. TSEPARAM outputs these parameters to LU4 in a form that TSE may use. TSEPARAM also sets the default values of any null parameters.

*All memory sizes in bytes.

The parameters are passed via the START command as discussed in Chapter 5 of Reference 2. The order of these parameters is as follows:

```
START , DISP, BGNA, ENDA, ELEV1, ELEV2, MODE, OLDATA
```

where the meaning and default values of each parameter are discussed ully in Section 3.2.1.

The only message printed by TSEPARAM is in the event of an I/O error on LU-4:

```
TSEPAR I/O ERR XXXX
TASK PAUSED
```

where XXXX is the returned status of the I/O driver as defined in Reference 8. The user may then either cancel the task, or continue using the CANCEL or CONTINUE commands as discussed in Reference 3. If TSEPARAM is continued after an I/O error, it will make another attempt to write to LU-4.

When TSEPARAM has successfully completed,

```
.BG:   END OF TASK 0
```

will be printed on the system console.

5.2.2 TSEMAIN.FTN

TSEMAIN is, as the name suggests, the main program of TSE. Its basic function is to simply call the different processing routines in their proper order. It also reads in the parameters set up in TSEPARAM from LU-4, sets the flag INDCTR if it is a sector scan, and utilizes the system clock to accommodate the time mode option.

The only error message printed by TSEMAIN is in the event of an I/O error on LU-4:

```
XXXX I/O ERROR IN RTN TSEMAIN
TASK PAUSED
```

where XXXX is the status returned by the I/O driver. If the user continues execution with the TPA command, TSEMAIN will attempt to read again.

When TSE has been successfully completed, TSEMAIN will either enter a time-of-day wait (SVC2, 10), pause and wait for a TCO command, or immediately begin again, according to the MODE parameter on the TSE command.

Reference 8. OS/32 Series General Purpose Driver Manual, Interdata Publication Number B29-384R04.

The first time TSE is run, TSEMAIN calls RDHIST(0) which reads the track history file from LU-6. Subsequently, after each run of TSE, TSEMAIN calls RDHIST(1), which writes the updated track history to LU-6.

TSEMAIN also monitors a second set of parameters in the task common EXTRA. When the CHANGE flag goes high, TSEMAIN changes the parameters which were specified during execution of the CHANGE task. To eliminate confusion of data, these changes are only implemented upon the completion of each run.

5.2.3 TSEHIST.FTN

TSEHIST contains the subroutine RDHIST, which is called by TSEMAIN. As its name suggests, RDHIST is involved with the history file, LU-6. When its argument is a 0, RDHIST reads the old track history from LU-6, and stores it in the task commons TASK and EXTRA. If there is no track history (parameter HIST in the TSE command is null), the track history pointers IPNT and NPNT are set to 0 and RDHIST returns.

When the argument of RDHIST is a one, RDHIST writes the present track history, found in task commons TASK and EXTRA, to LU-6 and returns.

No messages are printed by TSEHIST.

5.2.4 TSEDATA.FTN

TSEDATA contains the PPRDSC subroutine called by TSEMAIN. Its function is to input data from the PPP decoder and transfer it to the disc. Since FORTRAN is inherently slow for I/O, the actual input from the decoder had to be done in assembler. Therefore, although TSEDATA is a FORTRAN file, it actually contains very little FORTRAN.

PPRDSC has a single argument, a 6 element INTEGER *2 array. This array contains the first 6 parameters of the TSE command. The first and last elements are ignored. Elements 2 and 3 are the Begin and End angles of a sector scan. Elements 4 and 5 are ELEV1 and ELEV2 of the TSE command.

If the sector scan flag INDCTR is set to a 1, PPRDSC determines the direction of the scan according to Section 3.2.1.1. It also detects sector scans which cross 0^{o}, a potential hazard, and sets a flag (STPFLG) accordingly.

As noted earlier, there are two modes of input: sector scans anc PPIs. These will be discussed separately in the following sections.

5.2.4.1 Sector Scans

After determining the direction in which it wants to collect data, PPRDSC then determines which direction the radar is actually moving by subtracting two consecutive azimuth angles and checking the sign of the result. If the radar is moving in the wrong direction, PPRDSC waits until it is going in the right direction before continuing.

Once it has been determined that the radar is going in the right direction, PPRDSC compares the current azimuth to BGNA and ENDA. If the azimuth is within BGNA and ENDA, it waits until the azimuth is outside of the sector scan limits. This is to ensure that a complete scan is made.

Once the azimuth is outside of the sector scan limits, and is going in the right direction, PPRDSC waits until the angle is once again within the scan limits. At this point, if the current elevation is between ELEV1 and ELEV2, PPRDSC begins to collect data. As soon as the azimuth once again goes outside of the sector scan limits, PPRDSC stops collecting data, and flags the current azimuth as the last. PPRDSC no longer looks at elevation once it has started collecting data.

In order to be sure that no data is missed, a double buffering technique is used for data collection. With this method, while one azimuth is being written onto the disc, another is being read into a second buffer. In this way, the input and output are effectively accomplished simultaneously.

5.2.4.2 Full Scan

Once it has been determined that a full 360 degree scan is to be collected, PPRDSC reads in the current elevation angle of the radar. If this elevation angle is outside the upper and lower limits set by ELEV2 and ELEV1 in the TSE command, PPRDSC waits until it is within the set limits.

Once the elevation of the radar is satisfactory, PPRDSC saves the azimuth in R2, and begins collecting data. When it determines that the radar has once again passed the azimuth at which data collection began, PPRDSC flags the last azimuth and returns.

At the beginning of PPRDSC, the direction of the radar is tested by looking at the sign of the result when two consecutive azimuths are subtracted Once data collection has begun, PPRDSC assumes that the direction does not change, and therefore no longer checks it.

To expedite data collection, the double buffering technique discussed in Section 5.2.4.1 is used.

When PPRDSC is called by TSEMAIN, it logs "PPRDSC" to the console. The only other message printed by PPRDSC occurs when an I/O error is encountered, either from the PPRI or the DISC:

```
XXXX I/O ERROR
TASK PAUSED
```

where 'XXXX' is the status code returned from the I/O driver. If the TCO command is subsequently used, PPRDSC will start over from the beginning. All data which was previously collected will be lost, and new data will be input.

A common example of this is the message:

```
828A I/O ERROR
TASK PAUSED
```

This is the PPRI time-out error, which means that no data is getting to the PPRI from the PPP DECODER. If recorded data is being used, this could mean that the recorder came to the end of a track, or was turned off. The user can merely fix the problem and enter TCO, whereupon PPRDSC will begin collecting new data.

5.2.5 TSERANGE.FTN

RANGE, the range processor subroutine, does I/O and almost all its logic in assembly language. The first assembly language block unpacks ancillary data (except the azimuth angle and STOP, the last azimuth flag) from one of the buffers used in PPRDSC (COMMON/ZSTORE/). The second block unpacks the azimuth angle and the last-azimuth-flag and reduces the PPRI (Pulse Pair Recorder Interface) data to be compatible with the azimuth processor and computes the inner sums (Σ zr and $\Sigma z^2 r$). The subroutine loops through this second block for all azimuths.

The actual algorithm is rather simple: the processor examines each range cell in turn, computing sums and segment length upon acceptance. If the cell is not above the thresholds, its predecessor is considered the end of a segment. If the segment is longer than ρ cells, the segment and its sums are stored. Upon the acceptance of an eleventh segment, the shortest one is found and rejected. After each azimuth, the subroutine outputs segments and sums to COMMON blocks: /DATA/and/RUNSUM/.

The important variables used by RANGE are listed in the following table:

COMMON/MUSIG/PCTMIN	Percentage of range cells ignored because of ground clutter
MMU	Minimum acceptable absolute value of MEAN
SIGMA	Minimum acceptable variance
COMMON/EXTRA/RHO	Minimum acceptable segment length
GRND	Number of range cells ignored becuase of ground clutter
ZTH	Minimum acceptable power
BETA	Overlap constant used in AZPRO, not range
K	Running azimuth counter
COMMON/READZ/NRCEAD	End address for read, stored for use by CENTWT
OLDATA	Flag to tell program to ignore 9th bit of power
COMMON/ZSTORE/ANC(4)	Ancillary data in 32 bit words
ZEE(1024)	Video data in 32 bit words
COMMON/DATA/VAR(448, 10, 3)	Begin, end and label information used by other programs VAR(K, I, 1)= beginning of I-th segment on K-th azimuth VAR(K, I, 2)= end of Ith segment on K-th azimuth VAR(K, I, 3)=Label of I-th segment on the K-th azimuth (set to zero by RANGE)
COMMON/RUNSUM/ZZR(10, 448)	Σ zr
ZZR2(10, 448)	$\Sigma z^2 r/128$
TP	Range cell size $(75m*2^{TP})$
ELEVAT	Elevation
DAY, HOUR, MINUTE, SECOND	Self explanatory

5.2.6 TSEAZPRO.FTN

The azimuthal processor, TSEAZPRO, consists of five subroutines: MAKSEN, WAKBAK, LOOKBK, WIPOUT and WIPE. Its input is the VAR array from TSERANGE (begin, end, label). The labels are initially zero, but are assigned to azimuthally significant storm cells as TSEAZPRO proceeds. The VAR array is also the output from TSEAZPRO.

MAKSEN ("Make Sense") is the executive program. It sets the initial value of various variables and then calls WAKBAK ("Walk Back") for each segment. After all of the WAKBAK processing is complete, MAKSEN looks through the cross-reference array, XREF, and re-labels any segments which belong to equivalent cells.

The minimum acceptable number of adjacent azimuths, α, relates to the minimum sector length, ρ, and to the radius, r, of the center of the cell, as follows: $\alpha = \frac{\rho}{r}$ radians.
Since we know the average angular change, we can determine the minimum number of adjacent azimuths.

WAKBAK determines to which cell a given segment belongs, and wipes the cell out if it doesn't meet the minimum size requirement. It calls LOOKBK ("Look Back") and WIPOUT ("Wipe Out"). LOOKBK checks whether two cells are adjacent. The minimum overlap constant, β, defines adjacency. The end of one cell must be at least β range cells ahead of the beginning of the adjacent azimuth's cell. LOOKBK also checks whether either of the cells is already labelled. If it is, the program enters a number in the cross reference table, XREF, which makes them equivalent. If WAKBAK fails to count α adjacent cells, it calls WIPOUT which sets the VAR elements to (0, -1, 0).

The final outputs of TSEAZPRO are the VAR array, an array of good labels, LARRAY and the number of labels, NLAB.

5.2.7 TSECENT.FTN

TSECENT has two subroutines: CENTWT and ORDER. CENTWT determines, for each acceptable storm cell, the area, Z-weighted area, Z-weighted center, and the maxima of the power, mean and variance. ORDER is called to find and order the 12 cells with the largest Z-weighted area.

The following equations express the Z-weighted area, M, and the center location, $\overline{X}$; $\overline{Y}$:

$$M = \Sigma\Sigma z\,(r, \theta) r\, \Delta\, r\, \Delta\, \theta \qquad (5\text{-}1)$$

$$\overline{X} = \frac{1}{M}\ \Sigma\,\Sigma\, r^2\ \sin\,\theta\ z(r, \theta)\, \Delta\, r\, \Delta\, \theta \qquad (5\text{-}2)$$

$$\overline{Y} = \frac{1}{M}\ \Sigma\,\Sigma\, r^2\ \cos\,\theta\ z\,(r,\ \theta)\, \Delta\, r\, \Delta\, \theta \qquad (5\text{-}3)$$

TSERANGE calculated 2 inner sums, S_1 and S_2:

$$S_1(\theta) = \Sigma\ z(r, \theta)\ r\ \Delta\ r \qquad (5\text{-}4)$$

$$S_2(\theta) = \Sigma\ z(r, \theta)\ r^2\ \Delta\ r \qquad (5\text{-}5)$$

TSECENT uses these to get:

$$M = \Sigma\, S_1(\theta)\ \Delta\ \theta \qquad (5\text{-}6)$$

$$\overline{X} = \frac{1}{M}\ \Sigma\ \ S_2\ (\theta)\ \sin\,\theta\ \Delta\ \theta \qquad (5\text{-}7)$$

$$\overline{Y} = \frac{1}{M}\ \Sigma\, S_2\ (\theta)\ \cos\ \theta\ \Delta\ \theta \qquad (5\text{-}8)$$

TSECENT also calculates the unweighted area, A:

$$A = \Sigma \int_{r_1\,(\theta)}^{r_2(\theta)} r dr\ \Delta\theta = \Sigma\, \left(\frac{r_2^2 \ - \ r_1^2}{2} \right)\ \Delta\ \theta\ . \qquad (5\text{-}9)$$

where r_1 and r_2 are the beginning and end of a given segment.

TSECENT outputs the 12 largest sectors into a COMMON for use by TSEPLT.

5.2.8 TSEPLT.FTN

Subroutine TSEPLT takes the labels and centers from TSECENT and plots this information to the same scale as the SCRM data. Its argument, DISPLA, is the SCRM display number on which to plot. Most of the SCRM input or output is done in assembly language.

First TSEPLT clears the chosen display. It then reads the scales from one of the other displays. If there is no scaling information on any of the displays, the program will print "TURN ON DISPLAY AND CONTINUE" and then pause. After reading the scales, the program plots lines from the beginning to the end points of all of the strongest storm cells. We divide each azimuth interval into five steps and plot these lines at each. The last steps are to plot the range markers and, finally, the centers.

This last loop, the centers plot, stores away the centers, areas and maxima for all of the current centers. It plots the time history of the storm cell centers in their original colors. The earliest centers appear first and later ones cover the older ones. The program allocates storage for up to 24 past displays.

5.2.9 TSEDUMP.FTN

TSEDUMP is a FORTRAN program which runs concurrently with TSE, in its own partition of 5K bytes. Its function is to read the task commons which contain the track history data, and print the history in a useable form on LU-3. It runs with a priority of 11, just below the higher priority TSE task. This ensures that TSEDUMP will not interfere with efficient operation of TSE.

The printout is grouped, according to time, with the earliest times first, up to the latest data.

If no track history is yet available, TSEDUMP will print

```
NO HISTORY AVAILABLE
```

to the system console, and halt.

When the complete history has been printed out,

```
STOP
END OF TASK 0
```

will be logged to the console, and TSEDUMP will halt. TSEDUMP may be invoked by the DUMP command as discussed in Section 3.2.4.

5.2.10 TSECHNG.CAL

TSECHNG is an assembly language program which can be run concurrently with TSE, in its own partition. Its basic function is to allow the user to alter some of the program parameters without cancelling TSE and starting over. It accomplishes this by updating the task common EXTRA to notify TSEMAIN of the changes.

TSECHNG is invoked by the command

```
CHANGE
```

There are no parameters for this command. TSECHNG will prompt the user for input on the system console with

```
CHANGE > .
```

The user responds by typing the parameter to be changed, and the value to which it is to be altered. The possible parameters are shown in Table 5-1. Each parameter must be followed by an equal sign (=) and a decimal value. When all the parameters are the required values, CHANGE is terminated by typing

```
END.
```

This finalizes the changes and sets the CHANGE flag to notify TSEMAIN.

For example, if the power threshold is presently 150, and the user wishes it to be raised to 200, he simply enters the CHANGE command and waits for the prompt

```
CHANGE
*CHANGE>
```

He then types in

```
*CHANGE>  ZTH=200
*CHANGE> END
```

where *CHANGE> is the prompt and is printed by the system. The system will respond to END with

```
CHANGE: END OF TASK 0.
```

An unrecognizable input will cause CHANGE to print

```
CHANGE:  ILLEGAL COMMAND
CHANGE>
```

A parameter may be changed any number of times, but only the last change will be implemented.

PARAMETER	MEANING	POSSIBLE VALUES
RHO	RANGE THRESHOLD	$0 \leq x < NRC$
ZTH	POWER THRESHOLD	$0 \leq x < 512$
BETA	RANGE OVERLAP THRESHOLD	$0 \leq x < NRC$
PCTMIN	PERCENTAGE OF GROUND CLUTTER TO BE IGNORED	$0 \leq x < 100$
MMU	VELOCITY THRESHOLD	$0 \leq x < 256$
SIGMA	VARIANCE THRESHOLD	$0 \leq x < 256$
MINUTE	TIME BETWEEN DATA COLLECTION IN MINUTES	$0 \leq x < 1000$
DISPLAY	DISPLAY ON WHICH PLOT IS TO BE OUTPUT	$1 \leq x \leq 4$
BGNA	BEGINNING ANGLE FOR SECTOR SCAN, IF >360, A PPI WILL BE INPUT	$0 \leq x \leq 360$ $(x > 360 = PPI)$
ENDA	END ANGLE FOR SECTOR SCAN	$0 \leq x \leq 360$
ELEV1	LOWER ELEVATION THRESHOLD	$0 \leq x \leq 60$
ELEV2	UPPER ELEVATION THRESHOLD	$0 \leq x \leq 60$, Must be greater than ELEV1
MODE	MODE OF OPERATION	0 = TIME 1 = DEMAND 2 = CONTINUOUS
END	END OF CHANGE, STORE FLAG AND EXIT	

TABLE 5-1 OPTIONS FOR TSECHNG ROUTINE

APPENDIX A

TEST SOFTWARE FOR PPRI

APPENDIX A
TEST SOFTWARE FOR PPRI

The PPRI Test program is run as an executive task on the 7/32 computer, and therefore does not use the I/O drivers. The flow chart is shown in Figure A-1.

The program begins by outputting a CLR command and enabling the TEST mode on the PPRI. Now, by writing data to the PPRI, real data may be simulated for use in testing the interface. The test then outputs a SYNC1 code, and waits for an interrupt. If no interrupt is generated, an error is produced.

The second test performed by the program is to test the bit manipulation hardware by outputting ancillary and video test words and observing the input from the interface. For both cases, a word of all 1's is output, then all 0's, and then two cases of valid data. In the video mode, the data are chosen so that the x's sign magnitude/2's comp converter will be tested.

The program then outputs a sync code (not SYNC1) to be sure that sync codes are not passed to the FIFO. This test also ensures that ancillary data in subsequent subframes are ignored.

Since the input clock in the test mode is actually the DAG pulse from the processor, and due to the slow cycle time of the 7/32, the CPD cannot be tested in software.

There are 3 commands recognized by PPRI TEST:

S	Start test program
C	Continue test from last executed point (this allows the user to continue the test if an error is detected)
H	Halt the test and return to the OS. The return code is the number of errors encountered.

These commands are only processed when the program is in the command mode. They are ignored while the task is in progress. When the program is in command mode, an asterisk (*) is printed to the console.

PPRITEST is run by entering the CSS command PPRITEST. The format is

PPRITEST (DEV1:)

where DEV1: is the optional device to which messages are to be output. The valid devices are PAS: and CON:. The default is CON:, which means that all commands and messages are done on the system console. If for some reason the user wishes to use a CRT, TTY or DEC writer, he can connect it to the PASLA, and enter PAS: as the device. PPRITEST will then output all messages to the PASLA.

There are 6 error messages which may be output by PPRITEST. The messages, their meanings and possible causes are listed in Table A-1.

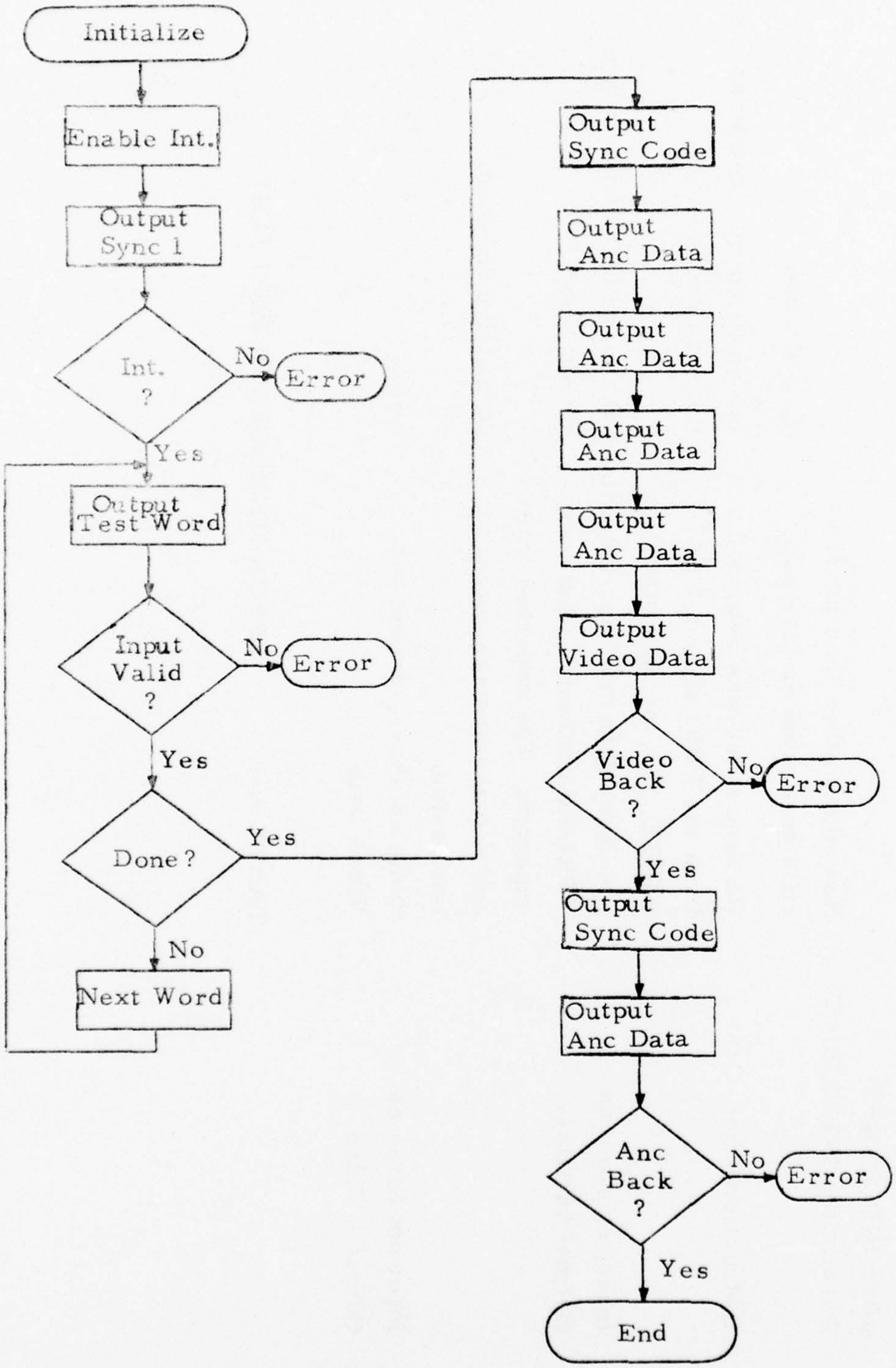


FIGURE A-1. FLOWCHART FOR PPRI TEST PROGRAM

	MSG	Reason	Probable Cause
1.	SYNC1 Code does not generate INTERRUPT		
2.	Invalid address returned after INTERRUPT		Sync Decoder
3.	Unsolicited INTERRUPT	Something other than a SYNC1. code generated an Interrupt	Sync Decoder
4.	Instruction Time-Out	No Sync pulse was returned from the PPRI after a READ/ WRITE, UC, or SS instruction	Interface not plugged in properly Address decode logic
5.	Input not same as Output-Ancillary	Ancillary data returned from the PPRI did not match what was expected. The expected input and actual input are output to list device.	Bit manipulation hardware Word Counter Output Synchronizer
6.	Input not Same as Output - Video	Same as above, except for Video Data	Same

TABLE A-1 ERROR MESSAGES IN PPRI TEST

APPENDIX B

SCRMI TEST SOFTWARE (SCRMITST)

APPENDIX B

SCRMI TEST SOFTWARE (SCRMITST)

Since the SCRMI is a transparent interface it can be tested with a modified version of Interdata's Common ULI test package (program number 06-129M96RO6A13). The modifications consist of adding an END command tc return to the operating system, and changing the interrupt service routines to be compatible with an OS/32-MT executive task. A complete description of the ULI Test Program may be found in Reference 9.

Unlike Interdata's test, the SCRMI test requires no jumper cable to connect the inputs and outputs. This connection is done on the SCRMI when the user outputs a TEST command. This is automatically done by SCRMITST.

There are four options recognized by SCRMITST. These are listed in Table B-1 and discussed in Reference 8.

The two commands recognized by SCRMITST are "RUN" and "END". When "RUN" is entered, the test is executed, and either a "NO ERRORS" or the appropriate error message will be printed on the list device. Unlike the ULI test, the SCRMI test does not loop on an error condition, but returns to command mode and prints an asterisk (*). To return to the operating system, type "END".

To run SCRMITST, use the CSS command SCRMITST. the format is:

```
SCRMITST     (DEV1:)
```

where DEV1: is the optional list device. If this parameter is null, or anything but PAS:, the system console (CON:) is used. If PAS: is entered, SCRMITST will do all command and message I/O to the pasla. The user may connect a TTY, CRT, or DECWRITER to the pasla for this purpose.

A list of error codes, reasons, and comments is given in Table B-2.

Reference 9. Common Universal Logic Interface Test.
Interdata PN. B06-129RO6, May 1975.

Option	Default	Description
DEVADR	X'8C'	SCRMI Interface Address
INTRPT	0	0 = Interface testing under interrupts 1 = No Interrupt Testing
INTLEV	0	Interrupt Level of Interface - Must be 0 for 7/32
MODE	0	0 = Byte Transfers 1 = Byte to Halfword Transfers

TABLE B-1 SCRMITST OPTIONS AND DEFAULT VALUES

Error No.	Reason	Comments

01	Halfword Mode Data Xfer	R4 = Data Written
		R6 = Data Read

02	Byte Mode of Byte to Halfword Mode Data Xfer	R4 = Data Written
		R6 = Data Read

03	Sense Status Bits (0:3)	R4 = Test Data
		R6 = Error Bits

04	Output Command Bits 4:7
	Or Sense Status Bits 4:7
		R4 = Output Command
		R6 = Returned Status

05	Unsolicited Interrupt	
06	No Interrupt	
07	Wrong Device Number Returned on Acknowledge	10 Bit Processor Only
08	Acknowledge Doesn't Reset Atn	16 Bit Processor Only
	Incorrect Interrupt Level = X	X = Register Set that the Interrupt Vectored Inot

TABLE B-2 ERROR CODES FOR SCRMITST

APPENDIX C

COMMON UTILIZATION IN TSE

Appendix C

COMMON UTILIZATION IN TSE

There are a total of eleven labelled common blocks in TSE. Four of these are task commons, and as such can be referenced by other tasks (e.g., TSEDUMP, or CHANGE). All common blocks used by the TSE system and their relative addresses are listed in Table C-1. Table C-2 lists the common utilization according to subroutine.

LOCAL COMMON BLOCKS:

ADDRESS	NAME
005CB8	MUSIG
005CC0	READZ
005CC8	SECTOR
005CD0	DATA
00C6A0	ALP
00C6A8	RUNSUM

TASK COMMON BLOCKS:

ADDRESS	NAME
020000	TASK
030000	ZSTORE
040000	DONE
050000	EXTRA

TABLE C-1 LIST OF COMMON BLOCKS AND THEIR RELATIVE ADDRESSES

Routine		
TSEMAIN	/MUSIG/PCTMIN, MMU, SIGMA	Parameters for TSERANGE
	/READZ/NRCEAD, OLD	Parameters for TSERANGE, TSECENT
	/SECTOR/INDCTR	Sector scan indicator for TSEDATA, TSEAZPRO, TSEPLT
	/EXTRA/RHO, GRND, ZTH, BG NA, K, RRAR EA(12, 24) RHO2, ZTH2, BETA2, PCT2, SIGMA2, MINUT2, ANG2(6), CHANGE	Parameters for TSERANGE and TSEAZPRC Also, area to be carried from TSEPLT to TSEHIST and TSGDUMP. The last eight variables are used by CHANGE
TSEDATA	/SECTOR/INDCTR	See TSEMAIN
	/ZSTORE/PPRI	First input buffer from PPRI, carried to TSERANGE
	/RUNSUM/PPRI2	Second input buffer, overwrites the running sum information
TSERANGE	/READZ/NRCEAD, OLDATA	See TSEMAIN
	/EXTRA/RHO, GRND, ZTH, BETA, K	See TSEMAIN
	/ZSTORE/ANC(4), ZEE(1024)	Data from PPRI or disc in same format
	/DATA/VAR(448, 10, 3)	Beginning, end, label info for good segmen carried to TSEAZPRO, TSECENT, and TSEPLT
	/RUNSUM/ZZR(10, 448), ZZR2(10, 448), AZIM(448), TP, ELEVAT, DAY, HOUR, MINUTE, SECOND	Information passed to TSECENT and TSEPI

TABLE C-2 (Page 1 of 3) TSE COMMON UTILIZATION

TSEAZPRO	/DATA/VAR	See TSERANGE
	/LBLS/LARRAY(50), NLAB	Good labels carried to TSECENT
	/DONE/XRF0, XREF(2050)	Cross reference table used between TSEAZPRO subroutines
	/ALP/ALPHFC	Azimuth/criterion used between TSEAZPRO subroutines
	/SECTOR/INDCTR	See TSEMAIN
	/EXTRA/RHO, GRND, ZTH, BETA, K	See TSEMAIN
TSECENT	/READZ/NRCEAD, OLDATA	See TSEMAIN
	/SECTOR/INDCTR	See TSEMAIN
	/ZSTORE/ANC(4), ZEE(1024)	See TSEMAIN
	/EXTRA/RHO, GRND, ZTH, BETA, K	See TSEMAIN
	/RUNSUM/ZZR(10, 448), ZZR2(10, 448), AZIM(448), TP, ELEVAT, DAY, HOUR, MINUTE, SECOND	See TSERANGE
	/DATA/VAR(448, 10, 3)	See TSERANGE
	/LBLS/LARRAY, I	See TSEAZPRO
	/DONE/FINAL, LFINAL, ARRAY, IPLOT, TWELVE, MAXFIN. MAX	Information carried to TSEPLT
TSEPLT	/DATA/BG(448, 10), EN(448, 10), LB(448, 10)	See TSERANGE
	/TASK/ZWAREA(12, 24), XCENT(12, 24), YCENT (12, 24), MAXPOW(12, 24), MAXVV(12, 24), TIME(24), IPNT, NPNT, DATE, ELEV(24)	Areas, centers, and maxima carried to TSEHIST and TSEDUMP
	/RUNSUM/PLTBUF(8960), AXIM(448), PP, ELEVAT, DAY, HOUR, MINUTE, SECOND	Information carried from TSERANGE. The plotting buffer overwrites the running sums which are no longer needed

TABLE C-2 (Page 2 of 3) TSE COMMON UTILIZATION

TSEPLT	(continued)	
	/SECTOR/INDCTR	See TSEMAIN
	/DONE/FINAL(4, 12), LFINAL(12), ARRAY(4, 50), IPLOT(3, 12), NUMLAB, MAXFIN(3, 12)	See TSECENT
	/EXTRA/RHO, GRND, ZTH, BETA, NAZ, RRAREA(12, 24)	See TSEMAIN
TSEHIST, TSEDUMP	/TASK/ZWAREA(12, 24), XCENT(12, 24), YCENT(12, 24), MAXPOW(12, 24), MAXVV(12, 24), TIME(24), IPNT, NPNT, DATE, ELEV(24)	SeeTSEPLT
	/EXTRA/I2DUM(5), RRAREA(12, 24)	See TSEMAIN

TABLE C-2 (Page 3 of 3) TSE COMMON UTILIZATION

APPENDIX D

ECHO TRACKING AND SIGNIFICANCE ESTIMATOR PHOTOGRAPHS

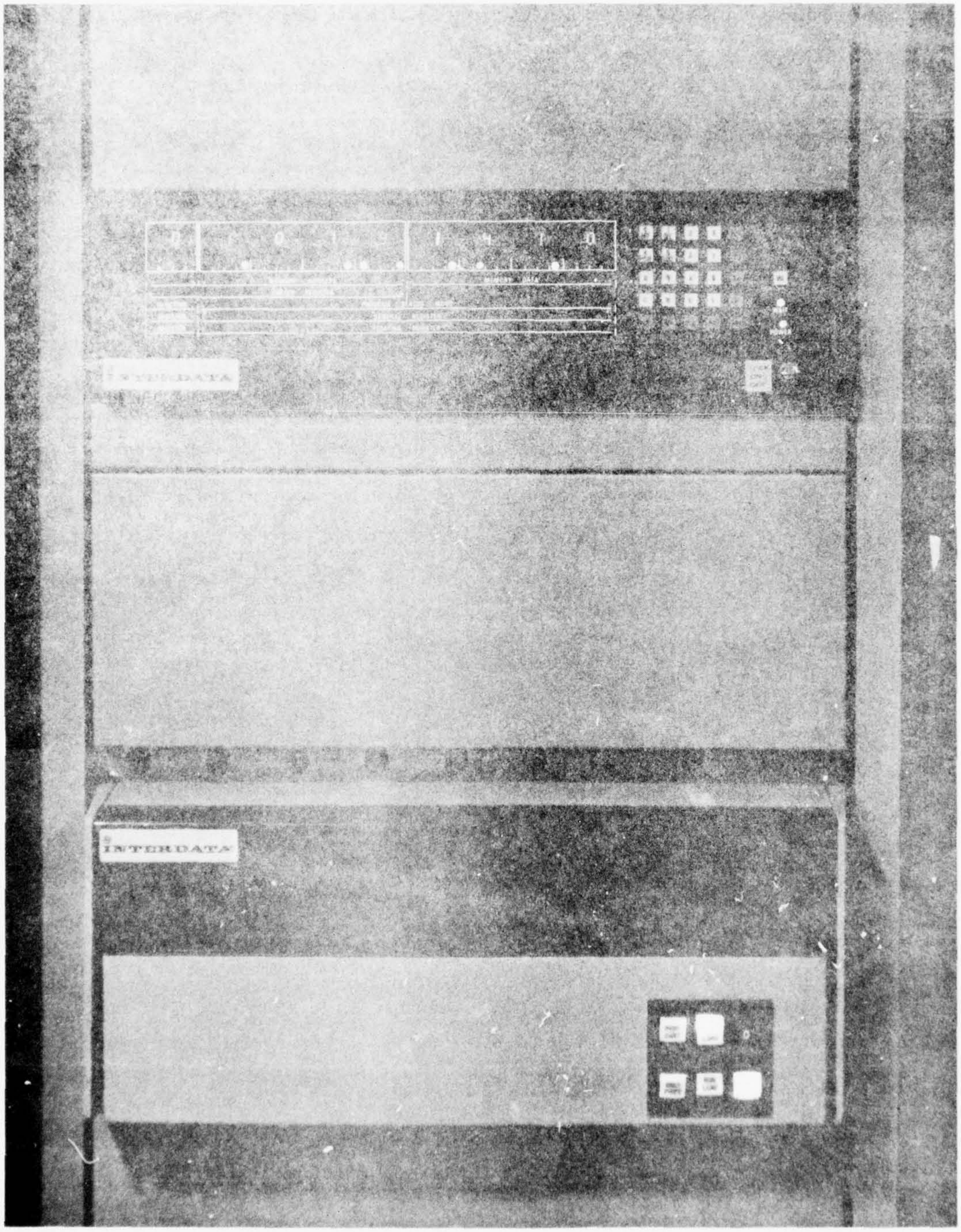


Figure D-1 Interdata 7/32 Processor and 10 M Byte Disc Drive

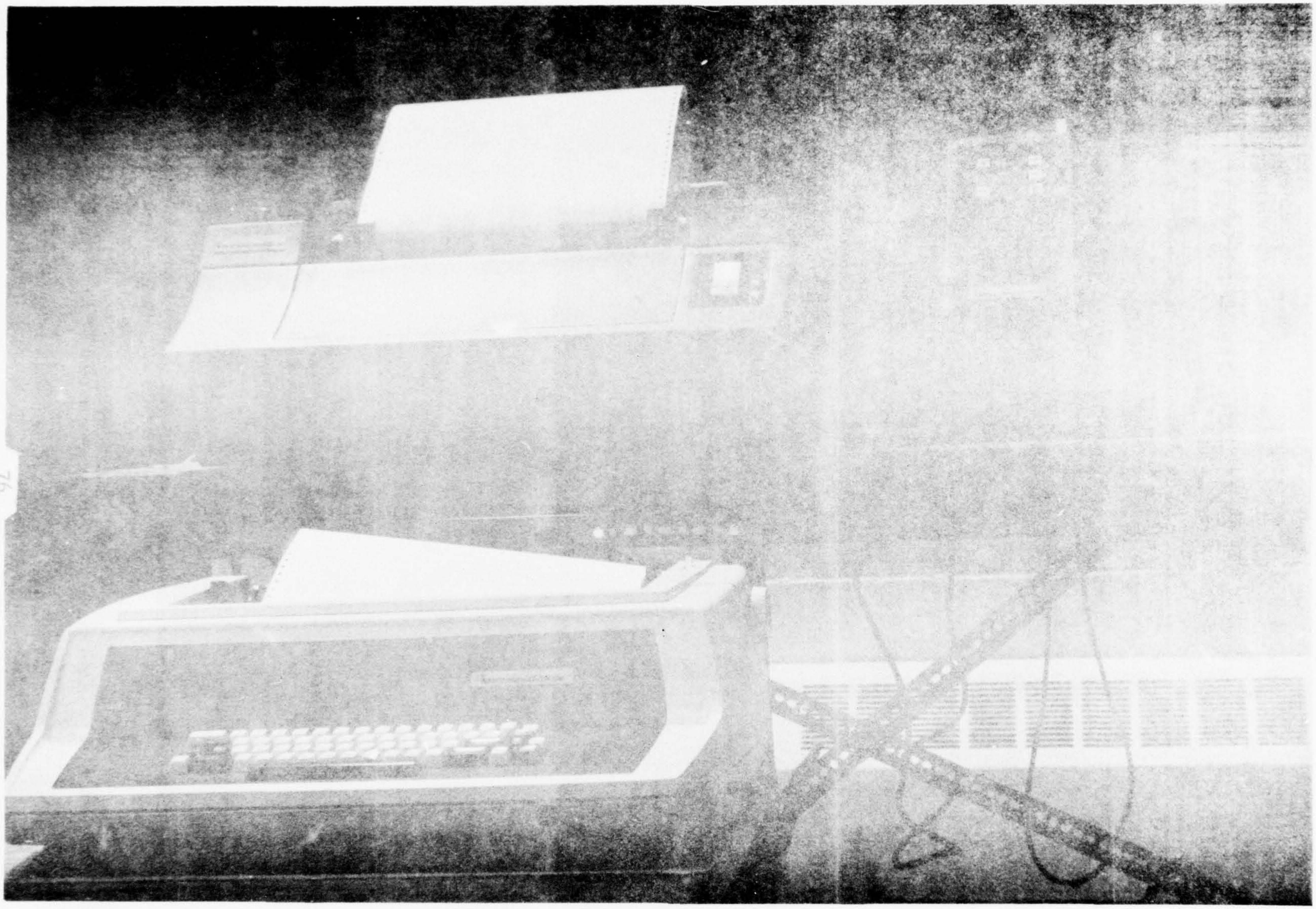


Figure D-2   Line Printer, Card Reader, and System Console

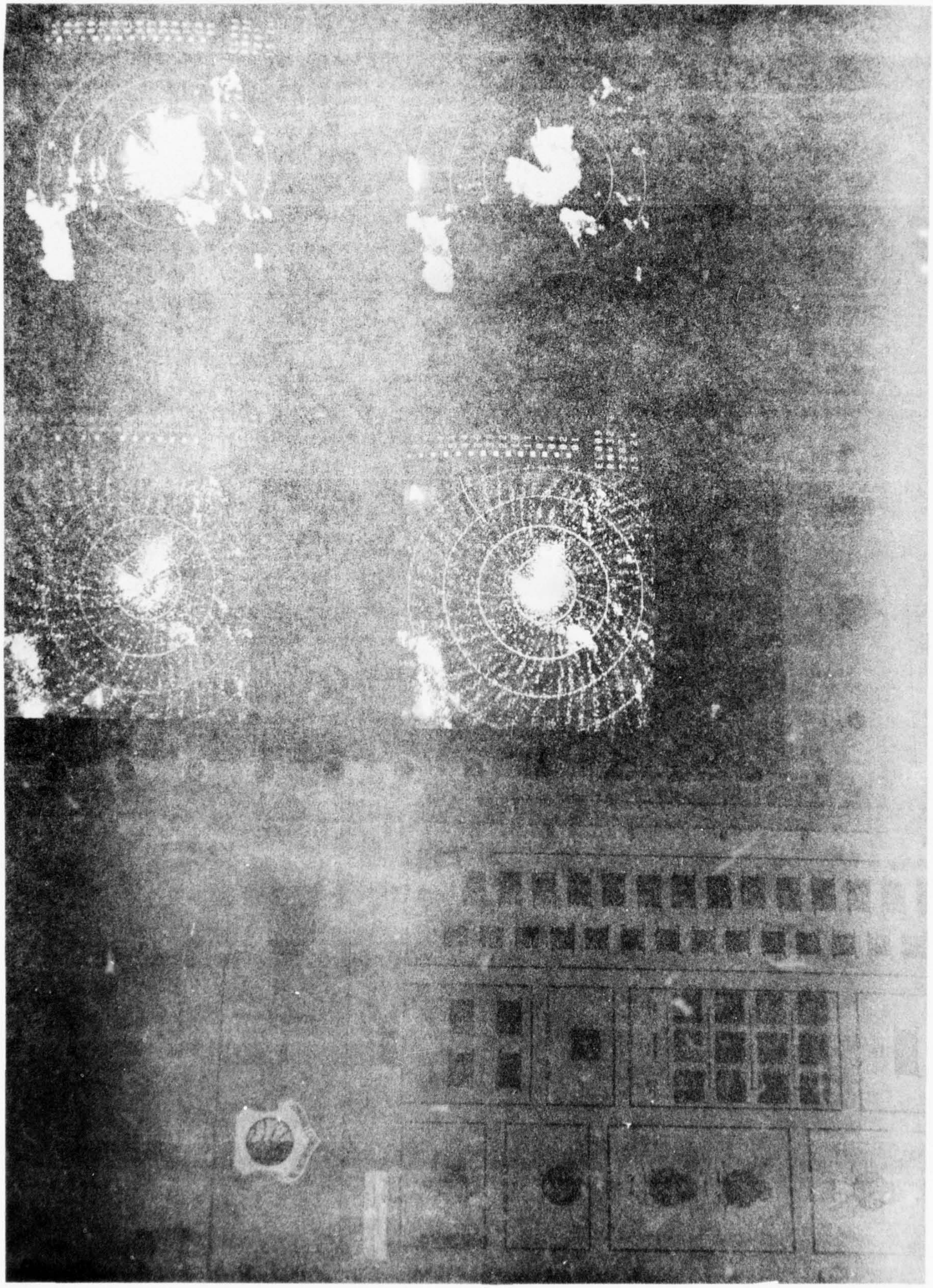


Figure D-3 Scan Converter/Refresh Memory

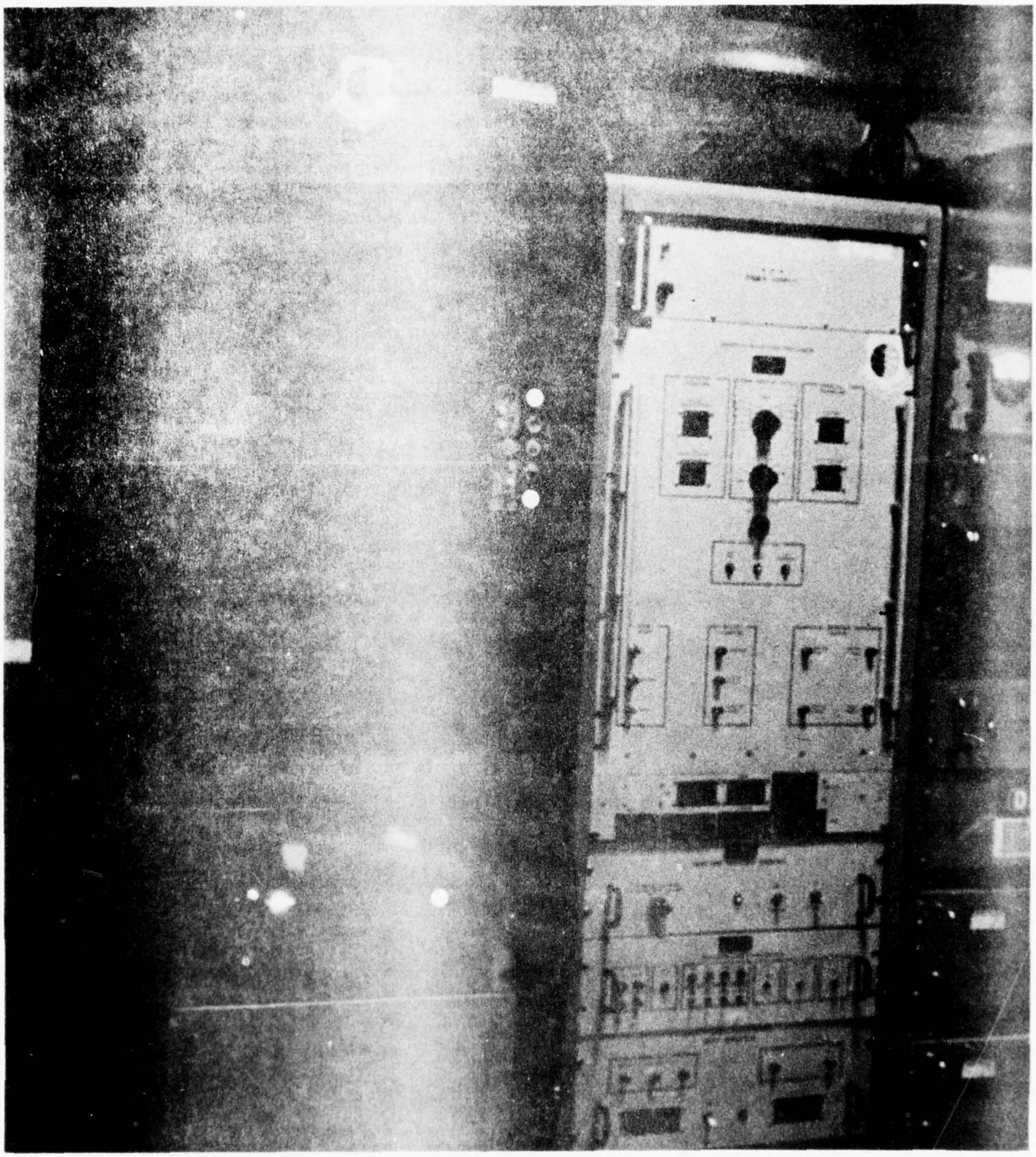


Figure D-4 Pulse Pair Processor, Encoder/Decoder (Right), and Tape Recorder (Left)

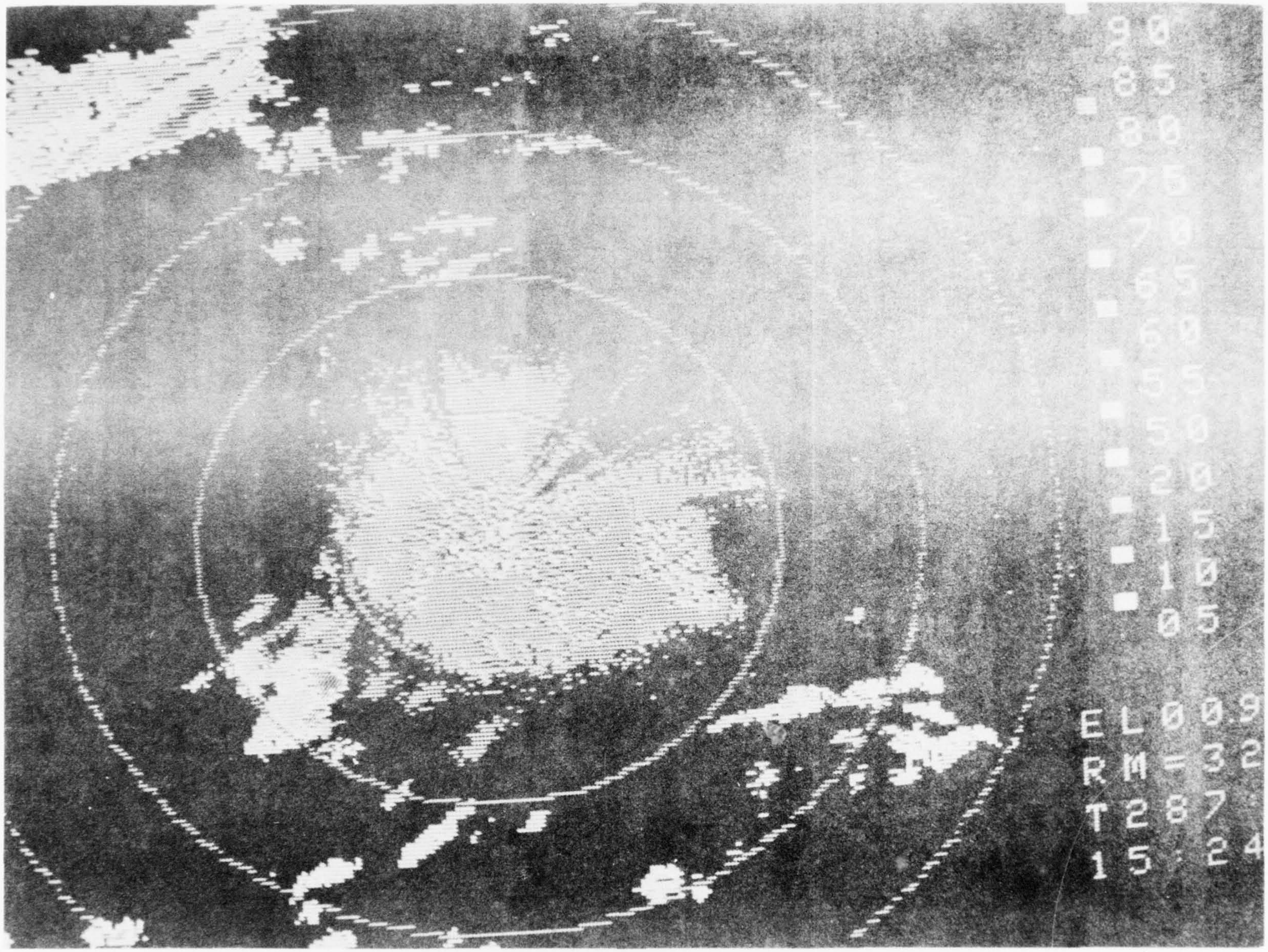


Figure D-5 Reflectivity Data

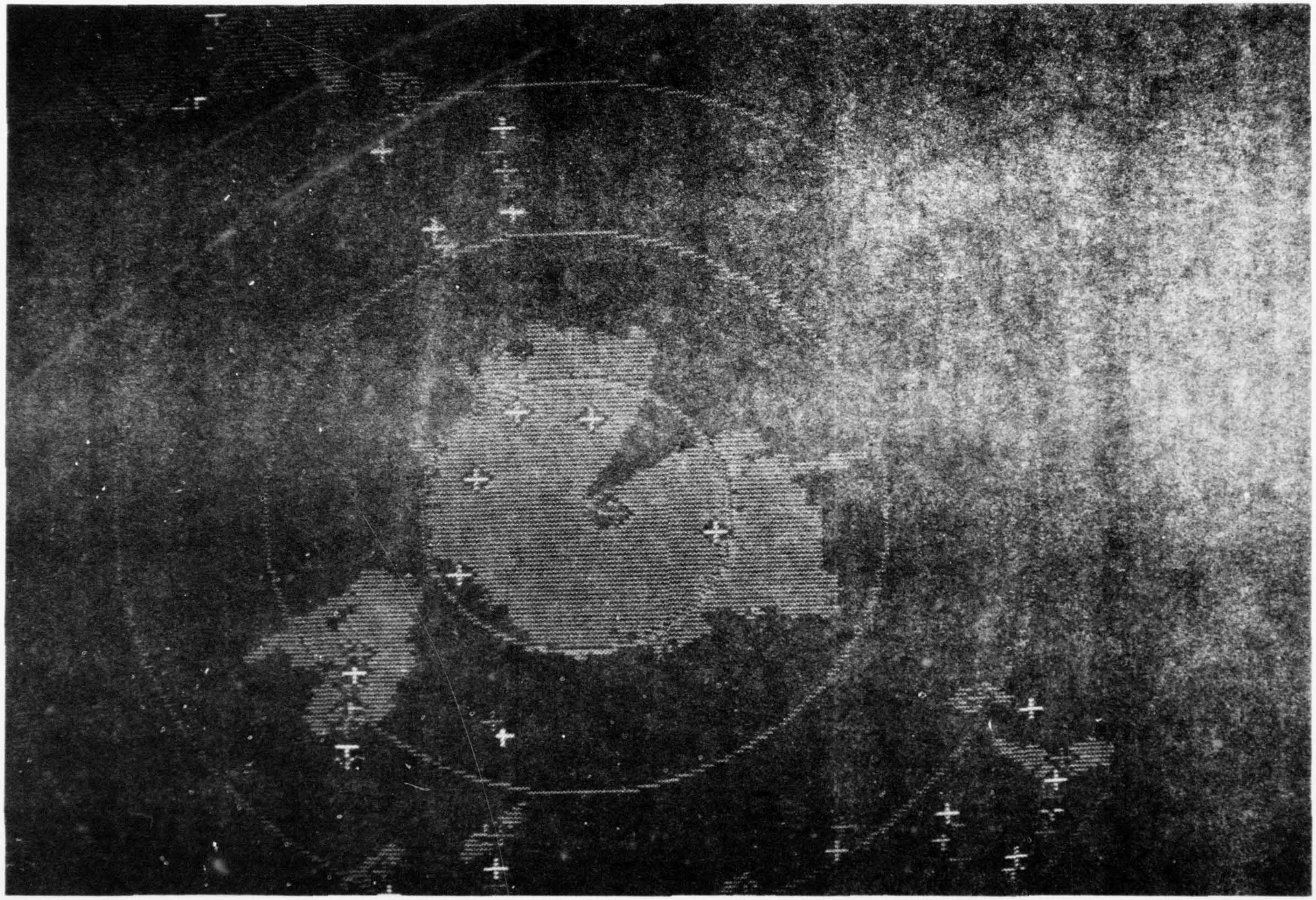


Figure D-6 TSE output on Data in Figure D-5. The cell at the center of the plot is local showers and ground clutter. The cell to the left of center is a storm cell moving northward, as evidenced by the past centers. Time between cell centers is five minutes.

APPENDIX E

LIST OF SCHEMATICS

APPENDIX E

LIST OF SCHEMATICS FOR THE ECHO TRACKING AND SIGNIFICANCE ESTIMATOR

Schematic	Size	# of Sheets	Drawing Number
Pulse Pair Recorder Interface	D	2	977973
Scan Converter/Refresh Memory Interface	D	1	977974
Cable Wiring Diagram: DDI to SCRMI	D	1	977975
PPP Recorder Decoder*	E	3	897817
PPP Recorder Encoder/Decoder Interconnection Diagram *	E	1	897650

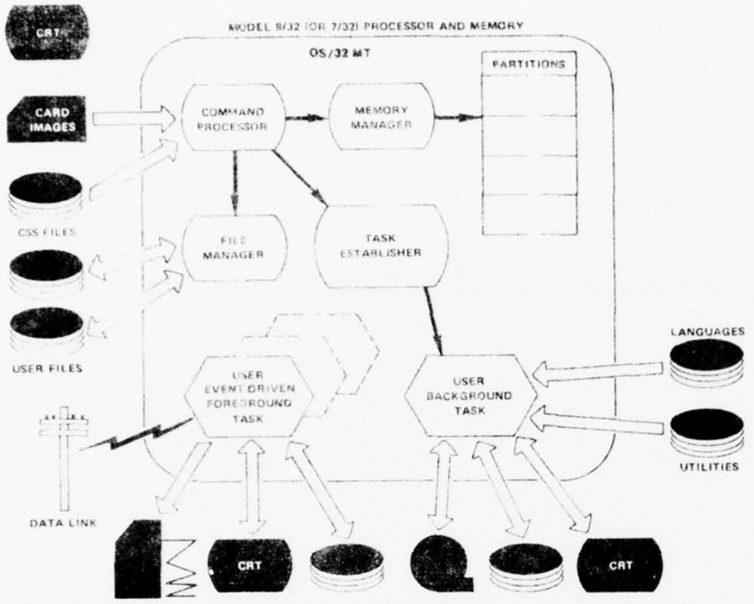
*Revised Drawings from earlier contracts

APPENDIX F

INTERDATA PRODUCT BULLETINS

INTERDATA

OPERATING SYSTEMS



OS/32 MT

GENERAL DESCRIPTION

OS/32 MT is an event-driven multitasking operating system for the Interdata 32-bit computer systems. The operating system provides concurrent support for foreground multitasking application systems and background batch oriented program development. The full power and flexibility of the Megamini architecture are made available to the user application system, by responsible yet responsive resource management facilities.

The program development features of OS/32 MT minimize the time and effort needed to test, debug and integrate user programs and systems. The powerful command language allows complex jobs to be performed easily with minimum operator intervention; the Command Substitution System allows use of parameterized catalogued command streams; automatic fault detection warns the user of problems as early as possible; Automatic Interactive Debugging (OS AIDS) is provided.

The OS/32 MT Filing System provides management of mass storage facilities ranging from the small 2.5 MB disc through multiple MSM discs. The file manager performs the tedious, yet complex housekeeping chores of space allocation, access coordination and static and dynamic file protection, allowing programmers to concentrate on making the best use of this valuable resource.

The device independent Input/Output system provides support for a wide range of peripheral devices, including unit record devices, magnetic tape, revolving mass storage, analog and digital equipment, and, through the use of ITAM, a wide range of communication equipment.

FEATURES

- Human Interface – powerful commands, catalogued job streams, foreground control with concurrent background operation.
- Resource Management – up to 255 tasks, up to 255 priorities, priority scheduling, time-slicing option, efficient allocation of memory space, automatic program relocation.
- Application System Support – Intertask communication and coordination, task-level interrupt handling traps, clock services with periodic facilities, multiple task commons and reentrant library, access to shared memory.
- File Management – Named files and devices, logical I/O, contiguous, chained and indexed file structures, random and sequential access, static and dynamic file protection.
- System Generation – Options as flexible as the hardware, software functions tailored to customer needs.

COMPREHENSIVE OPERATING FACILITIES

OS/32 MT gives comprehensive service to application programmers in all phases of their work, from the design through program preparation and development to operational support of completed applications. Furthermore, OS/32 MT provides these various functions concurrently, with user-defined priorities, so that on-line foreground systems can function securely while background program development is performed.

The major elements of the operating system are:

- Task Management
- Memory Management
- File Management
- Human Interface

TASK MANAGEMENT

The task management facilities provide all the functions required to create, schedule and execute an application system of cooperating user tasks. Foreground systems of over 250 tasks can operate with assured integrity concurrently with a background program development environment.

Task Scheduling is provided on a user-defined priority basis, with up to 255 separate task priorities. The priorities of tasks determine the order in which they gain control of the processor. Priority is also used to resolve input/output conflicts. Tasks at the same priority are serviced at the user's option, either first-come first-served or by time-sharing.

A foreground task is scheduled in response to one of four types of event:

- Operator Requests
- Hardware Interrupts
- Inter-Task Events
- Timer Service Events

OS/32 MT provides a wide range of intertask communication and coordination functions to foreground tasks. These enable the user to build well-structured systems of tasks. A supervisor call (SVC) allows a user task to:

- Load a task
- Suspend or release a task
- Start a task
- Cancel a task
- Queue a parameter to a task
- Send a block of data to a task
- Change a task's priority
- Respond to external interrupts

TASK-HANDLED TRAPS

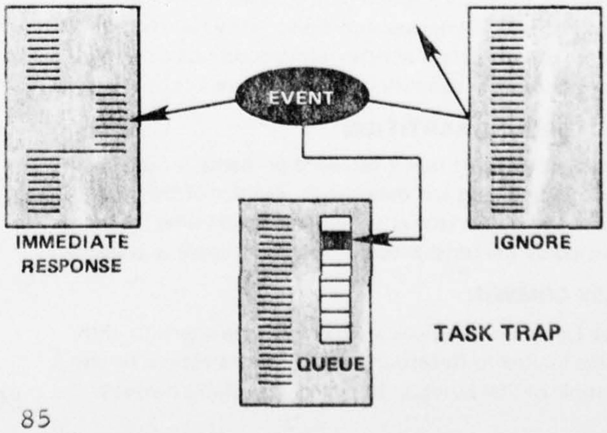
OS/32 MT provides a powerful interrupt handling capability at the task level – the Task-Handled Trap. This facility permits a task to be interrupted in its normal execution sequence by any of a variety of hardware and/or software conditions – particularly valuable for intertask communication. A task handled trap may occur for the following reasons:

- External interrupt from a trap generating device.
- Receipt of a message from another task.
- Completion of an I/O proceed request.
- Termination of a specified time delay.
- Communications with an OS/32 Debug System.
- Memory access fault.
- Illegal instruction, Arithmetic Fault
- Power restoration after outage

Task-handled trapping ennances the processing of multiple asynchronous events. For example, consider a task providing conversational facilities to a number of terminals simultaneously. By issuing I/O–proceed calls to all of the terminals and then entering Trap Wait, a task can respond to each transfer as it completes. It does not have to be continually scanning for transfers.

Task trapping also is useful when a task must be awakened in response to interrupts from some external device. Certain drivers, particularly the Eight-Line Interrupt Module, can cause a task trap in response to an interrupt from the device. The functions provided by OS/32 MT for handling these traps includes the complete ISA (Instrument Society of America) proposed standards for process control. These are:

- Connect – Attach a device to a task.
- Thaw – Enable interrupts on a device.
- SINT – Simulate an interrupt on a device. This extension to the standards is extremely useful for simulation purposes, and is frequently required for initialization and debug functions.
- Freeze – Disable interrupts on a device.
- Unconnect – Disconnect a device from a task.



TASK TRAP

MEMORY MANAGEMENT

The operating system enables a user to optimize the main memory structure for his application requirements. At system generation time, the user specifies the number and size of the foreground partitions. The programmer writes routines or subroutines without concern for physical addresses or memory reference format.

The CAL assembler produces object code optimized to take advantage of the various memory and time-saving short-form memory reference instructions of Interdata's 32-bit architecture. The details associated with address relocation are totally invisible to user tasks and are accomplished via the Memory Access Controller. Additionally, the Memory Access Controller allows loading of non-resident tasks directly into memory at DMA rates.

The OS itself and all static data structures (that is, those that are unchanging in size as time passes) are located in the lowest part of physical memory. Dynamic data structures, such as File Control Blocks, are located in the highest part of local memory. The remainder of physical memory is devoted to user-specified partitions.

MEMORY PROTECTION

A variety of memory protection mechanisms is provided by OS/32 MT.

Access to the system dynamic data is controlled so that no errant task can seize enough space to disturb the rest of the system. At task establishment a limit on the amount of system space is set for each task. It that limit is exceeded through some program error, the system call is rejected.

FOREGROUND PARTITIONS

More than 250 foreground partitions may be established at system generation. Optimal allocation of memory is achieved if partitions are arranged in ascending order of size with the smaller partitions at lower memory addresses.

A task may be dynamically loaded into any partition that is sufficiently large. No distinction is made between one partition and another. A particular partition need not be specified when a task is established or invoked. Therefore, tasks are automatically relocated to any available partition.

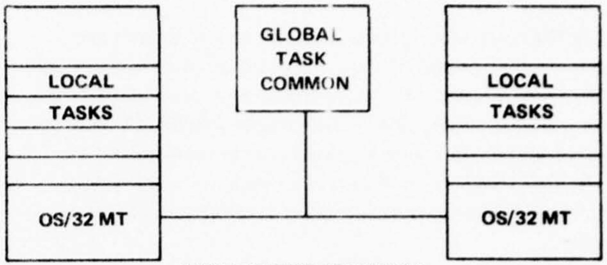
Tasks running in foreground partitions are permitted to perform a variety of intertask functions. They can request the loading and execution of other foreground tasks, cancel or delete tasks from memory, and pass parameters to tasks.

BACKGROUND PARTITION

A background partition is provided primarily for on-line program development and debugging. The size of the background is specified by the user at system generation time. It can be adjusted by the console operator as more space is available.

TASK COMMON

Task Common partitions are sharable data segments with access limited to foreground tasks. Their sizes can be any multiple of 256 bytes up to the total available memory. Task Common segments may also be positioned in multiported shared memory.



MULTIPORT MEMORY

RESIDENT LIBRARY

Programs shared by numerous tasks are maintained in the Resident Library, which can be any size from 256 bytes in 256-byte increments to 64 KB. Both foreground and background tasks may execute programs in the resident library.

TASK LOADING

Tasks are loaded in memory image so that the fastest possible response to a load request can be provided. All resolution of internal linkages and references to task common and the Resident Library are made at task establishment.

OVERLAYS

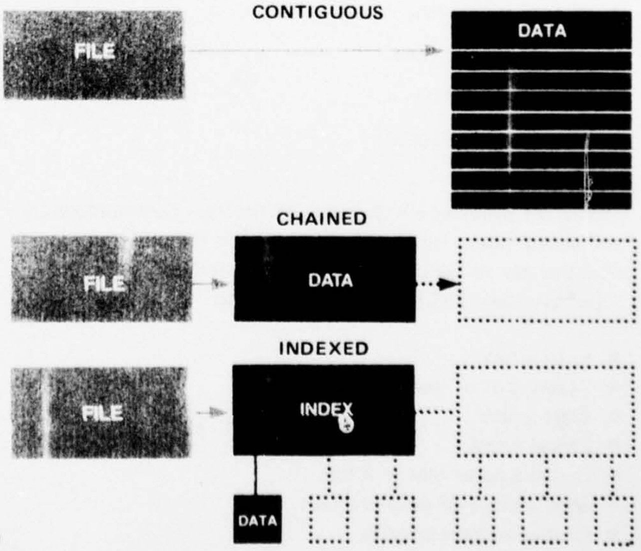
Tasks load overlays with a Supervisor Call. They are loaded in memory image with the minimum of system overhead.

FILE MANAGEMENT

Comprehensive file management is provided in the operating system. All discs are accessed through OS/32 MT's file manager. Three file structures, two access methods, and extensive protection features are provided.

FILE STRUCTURES

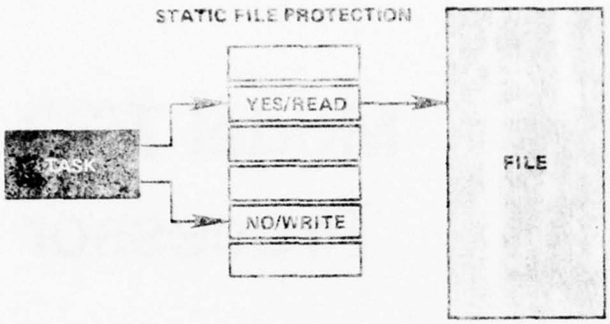
All three file types are served by the device-independent I/O system. Contiguous files provide unbuffered access to fixed sized user data structures. Chained files are optimized for sequential access to open ended data. Indexed files provide an ideal compromise between open-ended sequential access and fast random access.

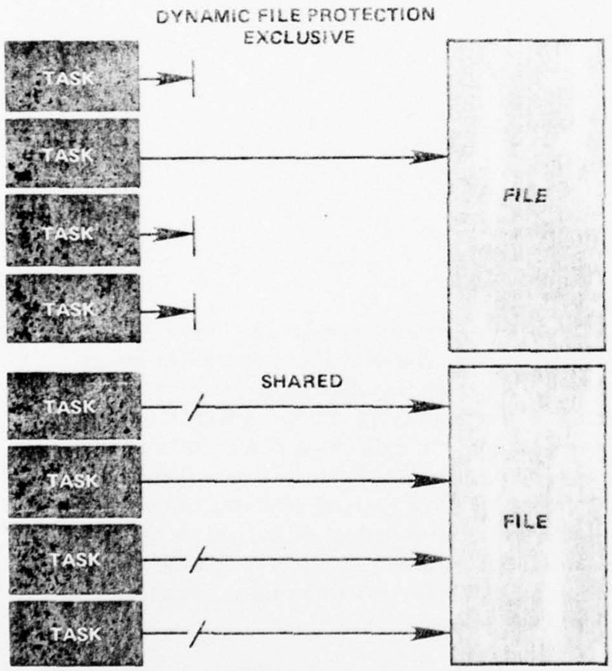


FILE PROTECTION

OS/32 MT provides two levels of protection: static and dynamic.

- Static file protection defines the fixed read-and-write access privilege associated with a file when it is created.
- Dynamic file protection specifies when the file is opened, whether the use will be shared or exclusive (for example, shared read – exclusive write).





HUMAN INTERFACE

The OS/32 MT man-machine interface is human engineered to simplify communication between the user and his system.

CONSOLE INTERFACE

The operating system is controlled by a console operator through a system console, typically a CRT, Carousel, or Teletypewriter. The operating system reads commands from the console and writes system messages to it.

The information contained herein is intended to be a general description and is subject to change with product enhancement.

Printed in U.S.A.

There are four classes of command:

- General System Commands
- Utility Commands
- Task-Related Commands
- Device and File Commands
- Command Substitution System

An extension to OS/32 MT, the Command Substitution System, allows commonly performed operations to be executed with one command.

The user establishes files of commands that are called from the console and executed in a defined sequence. In this way, complex operations can be carried out by the operator with only a small number of commands. These commands are analogous to macro-instructions in assembly language.

- A set of logical operators are provided to control the precise sequence of commands to be obeyed.
- Parameters can be passed to a CSS file so that general sequences can be written and will take on specific meaning only when the parameters are substituted.
- Other CSS can be created and called within the CSS file (nested commands).

SYSTEM REQUIREMENTS

MINIMUM HARDWARE

Model 7/32 or 8/32 processor
128KB memory
Display panel (Binary or Hex)
Power Fail/Auto Restart
Memory Access Controller
Operator Command Console (Note: If operator command console is a CRT then a binary input device or a Loader Storage Unit with an appropriate loader program is required for system bootstrap).
Universal Clock
Magnetic Media (9 Track Magnetic Tape, or 10 MB Disc).

SUPPORTED INTERDATA HARDWARE

Eight Line Interrupt Module
Line Printers (60-200, 300, and 600 lpm)
Card Reader (400 or 1000 cpm)
Paper Tape Reader/Punch
Cassette
Magnetic Tapes (9 Track 800 and 1600 BPI)
Discs (2.5 MB, 10 MB, 40 MB and MSM80, MSM300 storage modules)
Video Display (TTY Interface or 20 PASLA/PALS)
Loader Storage Unit
Carousel 30, 35, 300
Mini Input/Output System
Digital Multiplexor
Real Time Analog System
Note: OS/32 MT does not support peripheral devices interfaced with 10" (25.4 cm) device controllers.

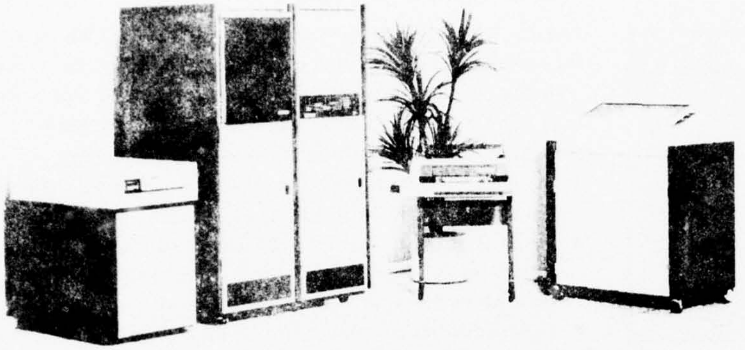
INTERDATA PRODUCT NUMBER

S90-006 OS/32MT

INTERDATA
A UNIT OF
PERKIN-ELMER DATA SYSTEMS
2 CRESCENT PLACE ● OCEANPORT, NEW JERSEY 07757
B761052

INTERDATA

PROCESSORS



Model 7/32 Processor

PRODUCT DESCRIPTION

The Interdata Model 7/32 Processor is a powerful, microprogrammed 32-bit word minicomputer that can directly address and support one million bytes of core memory. It features a large, comprehensive instruction set for bit, byte, halfword, and fullword operations as well as for fullword fixed-point arithmetic, list processing, data communications control, branching, and input/output.

The Model 7/32 is the entry-level system into Interdata's 32-bit computer line. Its performance is roughly equivalent to the IBM Systems 370/135.

The Model 7/32 was the industry's first 32-bit minicomputer. New options with the current 3-board Model 7/32-CII include High Speed Data Handling instructions and Single and Double Precision Floating Point Hardware.

FEATURES

- Two sets of 32-bit General Registers
- 16 Registers per set, 15 can be used for Indexing
- 32KB or 64KB Core Memory Modules
- 300 Nanosecond Access Time
- Cycle Time:
  750 or 1000 Nanoseconds for 32KB Modules
  1000 Nanoseconds for 64KB Modules
- Dual bus architecture

Up to eight memory modules, or 256K bytes, fit in the central processor chassis and operate as local memory. Memory above the first 256K bytes requires a Local Memory Bus Interface (LMBI) unit for each 256K bytes of memory added. In addition, an EDMA Buffer or Memory Access Controller (MAC) must be inserted between the DMA bus and LMBI to create the Extended DMA (EDMA) Bus, which allows peripheral devices and the central processor to address extended memory in the same way as local memory. MAC segments and relocates programs on 256-byte boundaries.

Extended memory access time is 800 nanoseconds. Cycle time is unchanged from local memory: ± 750 nanoseconds or 1000 nanoseconds depending on the core module used.

The 7/32's flexible I/O structure makes it easy to configure. The dual bus architecture supports 1023 I/O device addresses. The extended direct memory access (EDMA) bus can support 16 high speed devices on each of 7 selector channels. Devices on the multiplexor channel can use the auto driver channel to handle the housekeeping details associated with slow-speed character-by-character transfers.

Figure 1 shows a simplified block diagram of a 7/32 system configuration.

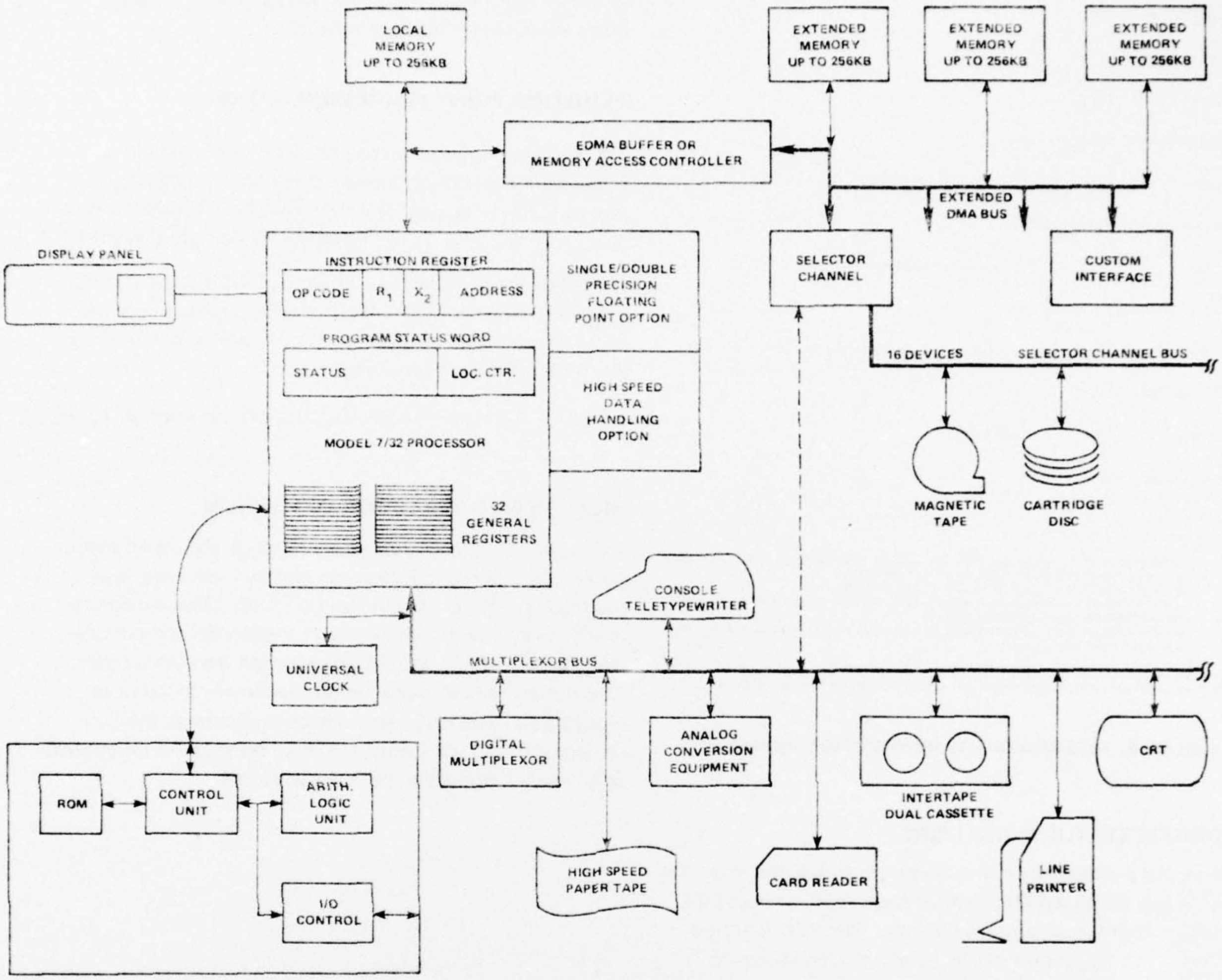


FIGURE 1. BLOCK DIAGRAM OF A 7/32 SYSTEM

ADDRESSING MEMORY

Seven instruction formats are implemented, as shown in Figure 2. Halfword formats are used for register-to-register or register with 4-bit constant operations. The fullword formats allow memory addressing to locations ± 16KB relative to the location counter or directly to 16KB. Both formats allow addressing a full megabyte of memory using indexing. The fullword format also allows specifying a 16-bit constant in the instruction.

A 48-bit instruction allows direct addressing of a full megabyte of memory. This 48-bit format also allows single and double indexing. Another 48-bit instruction format is also provided for specifying full 32-bit constants.

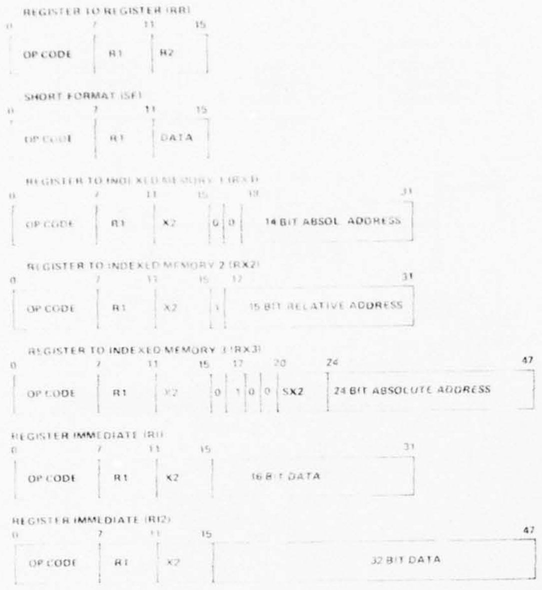
COMPATIBILITY

The 7/32 is peripheral- and data-compatible with all Interdata 16-bit and 32-bit processors. It is compatible at the language level with the Interdata 16-bit processors via MACRO CAL (Common Assembly Language) and FORTRAN V. The 7/32 is totally software compatible with the Interdata 8/32 Megamini.

The Model 7/32-CII supports the single- and double-precision hardware floating point option but will not support the single-precision firmware floating point option.

Single precision hardware floating point processor calculates a guard digit beyond the least significant digit in the result. If the hexadecimal guard digit is 8 or larger, the option rounds the least significant digit up. If the guard digit is less than 8, the least significant digit is unchanged.

The Interdata Model 7/32-CII does not implement the Halfword Mode of operation available on earlier Model 7/32 processors. The Model 7/32-CII is compatible with earlier Model 7/32 processors operating in 32-bit mode.



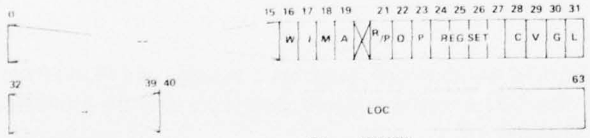
OP CODE HEXADECIMAL REPRESENTATION OF FUNCTION TO BE PERFORMED (ADD, MULTIPLY ETC.)
R1 SPECIFIES ONE OF 16 G.P. REGISTERS AS A FIRST OPERAND
R2 SPECIFIES ONE OF 16 G.P. REGISTERS AS A SECOND OPERAND
X2 SPECIFIES ONE OF 15 G.P. REGISTERS AS AN INDEX VALUE (ADD TO APPARENT ADDR SS FIELD TO OBTAIN TRUE VALUE OF ADDRESS)
SX2 SPECIFIES ONE OF 15 G.P. REGISTERS AS A SECOND INDEX VALUE (TO BE ADDED TO IM OF ADDRESS + FIRST INDEX VALUE)

A "RELATIVE" ADDRESS IS CALCULATED BY ADDING THE CURRENT LOCATION (ADDRESS) OF THE INSTRUCTION BEING EXECUTED TO THE SUM OF THE APPARENT ADDRESS AND THE INDEX VALUE

FIGURE 2. INTERDATA 7/32 INSTRUCTION FORMATS

PROGRAM STATUS WORD (PSW)

The central processor operates under control of the Program Status Word, a 64-bit field of flags, masks, select bits, conditi n codes, and location counter. The PSW is stored in memory as two 32-bit words. The bit designations in the PSW are shown in Figure 3.



BIT	MEANING	BIT	MEANING
0 15	RESERVED MUST BE ZERO	21	RELOCA ON/PROTECTION INTERRUPT MASK
16	WAIT STATE	22	SYSTEM UEUE SERVICE INTERRUPT MASK
17	IMMEDIATE INTERRUPT MASK	23	PROTECT MODE
18	MACHINE MALFUNCTION INTERRUPT MASK	24 27	REGISTER SET SELECT
19	ARITHMETIC FAULT INTERRUPT MASK	28 31	NDITION CODE
20	RESERVED MUST BE ZERO	32 39	RESERVED - MUST BE ZERO
		40 63	LOCATION COUNTER

FIGURE 3. INTERDATA 7/32 PROGRAM STATUS WORD (PSW)

LIST PROCESSING

The List Processing instructions manipulate entri s stored in slots in a conceptually circular list using param ters stored within the list. The parameters include the numb r of slots in the list, number of slots used, the current top nd bottom and the next top and bottom. List Proce sing instructions add entries to or remove entries fro n the top or bottom of the list, thus providing a facility f r push-down stacks and queuing algorithms.

FLOATING POINT PROCESSOR OPTION

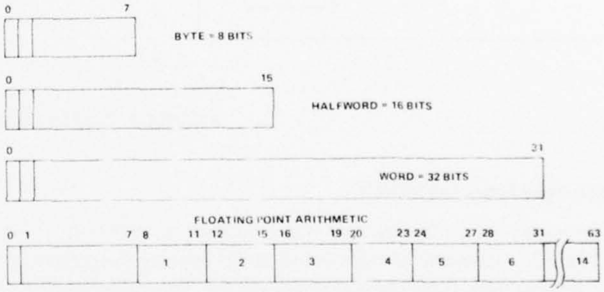
The hardwar floating point processor is available in a single/double precision model. The data formats, as shown in Figure 4, are totally compatible with the floating point forma s used on the Interdata Models 8/16 and 8/32.

The single/d uble p ecision floating point processor option includes two sets of eight floating point registers, one set for single-precision and one set for double-precision. Each set of registers is 64 bits long.

Times for the floating point instructions are listed in Table 1.

HIGH SPEED DATA HANDLING OPTION

This option provides facilities to comput the cyclic redundancy chec (CRC) polynomial used by r ost data communication protoc ls. In addition, it in ludes a memory-to-memory yte string move with transla ion and cumulative check e ther on the data byte or the ranslated byte. This option can calculate the Binary Synchronous Communication (BISYNC) error check polynomial, the Synchronous Data Link Control (SDLC) error check polynomial or any oth r defined error-check protocol.



BIT 0 = SIGN OF FRACTION
BITS 1 THROUGH 7 = EXPONENT OF HEXADECIMAL FRACTION EXPONENT EXPRESSED IN EXCESS 64 NOTATION.
BITS 8 THROUGH 31 = SINGLE PRECISION FRACTION OF 6 HEXADECIMAL DIGITS.
BITS 8 THROUGH 63 = DOUBLE PRECISION FRACTION OF 14 HEXADECIMAL DIGITS.

FIGURE 4. INTERDATA 7/32 DATA FORMATS FOR FIXED AND FLOATING POINT ARITHMETIC

TABLE 1. INTERDATA MODEL 7/32 INSTRUCTION EXECUTION TIMES

Type	Instruction	Execution Time in µsec*	
		750 nsec Memory	1000 nsec Memory
LOAD AND STORE INSTRUCTIONS	Load	3.25	4.00
	Load Register	1.00	1.00
	Load Immediate	2.50	3.00
	Load Immediate Short	1.00	1.00
	Load Complement Short	1.50	1.50
	Load Halfword	2.75	3.00
	Load Halfword Immediate	1.75	2.00
	Load Halfword Logical	2.50	3.25
	Load Address	2.00	2.25
	Load Multiple	2.50+2.00n	2.75+2.25n
	Store	3.25	4.00
	Store Halfword	2.50	3.00
	Store Multiple	2.50+1.50n	2.50+2.00n
	Exchange Halfword Register	1.00	1.00
		(n = number of words)	
FIXED POINT ARITHMETIC INSTRUCTIONS	Add	3.25	4.00
	Add Register	1.00	1.00
	Add Immediate	2.50	3.00
	Add Immediate Short	1.25	1.25
	Add Halfword	2.75	3.00
	Add Halfword Immediate	1.75	2.00
	Add to Memory	5.00	6.00
	Add Halfword to Memory	3.50	4.00
	Subtract	3.25	4.00
	Subtract Register	1.00	1.00
	Subtract Immediate	2.50	3.00
	Subtract Immediate Short	1.25	1.25
	Subtract Halfword	2.75	3.00
	Subtract Halfword Immediate	1.75	2.00
	Compare	4.00	4.50
	Compare Register	1.75	1.75
	Compare Immediate	3.25	3.75
	Compare Halfword	3.50	3.75
	Compare Halfword Immediate	2.50	2.75
	Multiply	24.00	24.50
	Multiply Register	21.75	21.75
	Multiply Halfword	6.00	6.25
	Multiply Halfword Register	4.25	4.25
	Divide	82.75	83.50
	Divide Register	80.25	80.25
	Divide Halfword	13.00	13.25
	Divide Halfword Register	11.00	11.00
FIXED POINT LOGICAL INSTRUCTIONS	AND	3.25	4.00
	AND Register	1.00	1.00
	AND Immediate	2.50	3.00
	AND Halfword	2.75	3.00
	AND Halfword Immediate	1.75	2.00
	OR	3.25	4.00
	OR Register	1.00	1.00

Type	Instruction	Execution Time in µsec*	
		750 nsec Memory	1000 nsec Memory
FIXED POINT LOGICAL INSTRUCTIONS (Cont.)	OR Immediate	2.50	3.00
	OR Halfword	2.75	3.00
	OR Halfword Immediate	1.75	2.00
	Exclusive OR	3.25	4.00
	Exclusive OR Register	1.00	1.00
	Exclusive OR Immediate	2.50	3.00
	Exclusive OR Halfword	2.75	3.00
	Exclusive OR Halfword Immediate	1.75	2.00
	Compare Logical	3.25	4.00
	Compare Logical Register	1.00	1.00
	Compare Logical Immediate	2.50	3.00
	Compare Logical Halfword	2.75	3.00
	Compare Logical Halfword Immediate	1.75	2.00
	Test Immediate	2.50	3.00
	Test Halfword Immediate	1.75	2.00
	Test and Set	3.25	3.75
SHIFT INSTRUCTIONS	Shift Right Logical	3.25+.25 [(n-2)/2]	3.50+.25 [(n-2)/2]
	Shift Right Halfword Logical Short	1.75+.25 (n-1)	1.75+.25 (n-1)
	Shift Right Halfword Logical	2.00+.25 (n-1)	2.25+.25 (n-1)
	Shift Right Logical Short	2.75+.25 [(n-2)/2]	2.75+.25 [(n-2)/2]
	Shift Left Logical	3.25+.25 [(n-2)/2]	3.50+.25 [(n-2)/2]
	Shift Left Halfword Logical Short	1.75+.25 (n-1)	1.75+.25 (n-1)
	Shift Left Halfword Logical	2.00+.25 (n-1)	2.25+.25 (n-1)
	Shift Left Logical Short	2.75+.25 [(n-2)/2]	2.75+.25 [(n-2)/2]
	Shift Right Arithmetic	3.75+.25 [(n-2)/2]	4.00+.25 [(n-2)/2]
	Shift Right Halfword Arithmetic	2.50+.25 (n-1)	2.75+.25 (n-1)
	Shift Left Arithmetic	3.75+.25 [(n-2)/2]	4.00+.25 [(n-2)/2]
	Shift Left Halfword Arithmetic	2.75+.25 (n-1)	3.00+.25 (n-1)
	Rotate Right Logical	1.75+1.00n	2.00+1.00n
	Rotate Left Logical	1.75+1.00n	2.00+1.00n
		(n = number of bits shifted)	
FLOATING POINT INSTRUCTIONS (HARDWARE)	Add	6.00	6.50
	Add Double	9.75	10.75
	Add Register	3.75	3.75
	Add Double Register	3.75	3.75
	Subtract	6.00	6.50
	Subtract Double	9.75	11.75
	Subtract Register	3.75	3.75

TABLE 1. INTERDATA MODEL 7/32 INSTRUCTION EXECUTION TIMES (Continued)

Type	Instruction	Execution Time in μsec*	
		750 nsec Memory	1000 nsec Memory
FLOATING POINT INSTRUCTIONS (HARDWARE) (Cont.)	Subtract Double Register	3.75	3.75
	Compare	6.00	6.50
	Compare Double	8.50	9.50
	Compare Register	3.00	3.00
	Compare Double Register	3.00	3.00
	Multiply	12.75	13.25
	Multiply Double	21.25	22.25
	Multiply Register	10.50	10.50
	Multiply Double Register	16.25	16.25
	Divide	13.25	13.75
	Divide Double	22.00	23.00
	Divide Register	10.75	10.75
	Divide Double Register	16.75	16.75
	Load	5.25	5.75
	Load Double	7.75	8.75
	Load Register	2.00	2.00
	Load Double Register	2.00	2.00
	Load Multiple	3.25+3.00n	3.50+3.25n
		(n = number of 32-bit operands)	
	Store	4.50	5.00
	Store Double	6.75	7.75
	Store Multiple	3.25+2.75n	3.50+2.75n
		(n = number of 32-bit operands)	
	Fix	6.00	6.50
	Fix Double	6.00	6.50
	Float	4.50	5.00
	Float Double	4.50	5.00
BYTE HANDLING INSTRUCTIONS	Load Byte	2.75	3.25
	Load Byte Register	1.25	1.25
	Store Byte	3.25	4.00
	Store Byte Register	2.00	2.00
	Exchange Byte Register	1.00	1.00
	Compare Logical Byte	3.00	3.25
BRANCH INSTRUCTIONS	Branch on True Condition	2.00	2.25
	Branch on True Condition Register	1.50	1.50
	Branch on True Condition Forward Short	1.50	1.50
	Branch on True Condition Backward Short	1.50	1.50
	Branch on False Condition	2.00	2.25
	Branch on False Condition Register	1.50	1.50
	Branch on False Condition Forward Short	1.50	1.50
	Branch on False Condition Backward Short	1.50	1.50
	Branch and Link	2.00	2.25

Type	Instruction	Execution Time in μsec*	
		750 nsec Memory	1000 nsec Memory
BRANCH INSTRUCTIONS (Cont.)	Branch and Link Register	1.50	1.50
	Branch on Index High	4.75	5.00
	Branch on Index Low or Equal	4.75	5.00
	Convert to Halfword Value Register	2.75	2.75
COMMUNICATIONS INSTRUCTIONS	Cyclic Redundancy Check 12	11.50	12.00
	Cyclic Redundancy Check 16	13.00	13.50
	Translate	4.25	5.00
	Simulate Channel Program	5.50	5.75
BIT MANIPULATION INSTRUCTIONS	Test Bit	5.75	6.00
	Set Bit	6.25	6.50
	Reset Bit	6.00	6.25
	Complement Bit	6.25	6.50
STATUS AND CONTROL INSTRUCTIONS	Load Program Status Word	6.50	7.50
	Load Program Status Word Register	3.50	3.50
	Exchange Program Status Register	3.75	3.75
	Supervisor Call	7.00	7.50
LIST HANDLING INSTRUCTIONS	Add to Top of List	4 5	5.25
	Add to Bottom of List	4.75	5.25
	Remove From Top of List	5.00	5.50
	Remove From Bottom of List	5.00	5.50
DATA HANDLING INSTRUCTIONS	Process Byte	5.00	5.25
	Process Byte Register	3.50	3.50
	Move and Process Byte String Register	(See Below)	*(See* Below)
INPUT/OUTPUT INSTRUCTIONS	Autoload	6.24+2.50 L+2.50n	6.50+2.50 L+2.50n
	Simulate Interrupt	7.50	7.75
	Read Data	4.00	4.25
		(L = length of leader in bytes, n = number of bytes)	

TABLE 1. INTERDATA MODEL 7/32 INSTRUCTION EXECUTION TIMES (Continued)

		Execution Time in µsec*				Execution Time in µsec*	
Type	Instruction	750 nsec Memory	1000 nsec Memory	Type	Instruction	750 nsec Memory	1000 nsec Memory
INPUT/OUTPUT INSTRUCTIONS (Cont.)	Read Data Register	2.25	2.25	INPUT/OUTPUT INSTRUCTIONS (Cont.)	Write Block Register	4.50+2.75n	4.50+2.75n
	Read Halfword	4.00	4.25		Sense Status	4.50	5.00
	Read Halfword Register	2.25	2.25		Sense Status Register	3.00	3.00
	Read Block	7.25+2.50n	7.50+2.50n		Output Command	4.25	4.50
	Read Block Register	4.50+2.50n	4.50+2.50n		Output Command Register	3.75	3.75
	Write Data	3.75	4.00		Auto Driver Channel	(See Below)	(See Below)
	Write Data Register	2.25	2.25				
	Write Halfword	3.50	3.75			(n = number of bytes)	
	Write Halfword Register	2.50	2.50				
	Write Block	7.25+2.75n	7.50+2.75n				

MOVE AND PROCESS BYTE STRING REGISTER

Execution Time in µsec

750 Nanosecond Memory

Sequence	Count Minus	First Byte	Next Byte	Special Character	Interrupted Byte	Last Byte
Error Checking Only	3.75	8.25	4.25	---	3.00	6.25
Translate Only	3.75	7.75	4.75	5.75	1.50	5.75
Check Then Translate	3.75	9.50	5.25	7.25	2.75	7.25
Translate Then Check	3.75	9.25	5.25	7.25	2.75	6.75

1000 Nanosecond Memory

Sequence	Count Minus	First Byte	Next Byte	Special Character	Interrupted Byte	Last Byte
Error Checking Only	3.75	8.50	4.50	---	3.00	6.50
Translate Only	3.75	8.00	5.00	6.00	1.50	6.00
Check Then Translate	3.75	9.75	5.50	7.25	2.75	7.50
Translate Then Check	3.75	9.50	5.50	7.50	2.75	7.00

AUTO DRIVER CHANNEL

Execution Time in µsec

750 Nanosecond Memory

Sequence	No Execute	Bad Status	Count Plus	Normal	Buffered End
Read (Byte)	10.00	11.25	12.50	16.25	16.50
Read (Halfword)	10.00	11.25	12.50	16.75	17.00
Write (Byte)	10.00	11.25	12.50	16.25	16.50
Write (Halfword)	10.00	11.25	12.50	17.50	17.75

1000 Nanosecond Memory

Sequence	No Execute	Bad Status	Count Plus	Normal	Buffered End
Read (Byte)	10.00	11.25	12.75	17.00	17.25
Read (Halfword)	10.00	11.25	12.75	17.50	17.75
Write (Byte)	10.00	11.25	12.75	17.00	17.25
Write (Halfword)	10.00	11.25	12.75	18.25	18.50

*Minimal Time Only (Partial Listing Only)

SOFTWARE

Software for the 7/32 runs under control of the OS/32 MT Multitasking operating system. Functional units within OS/32 MT provide facilities for user program development and execution.

- Task Management
- Memory Management
- File Management
- Man-Machine Interface

Languages supported for program development include the following:

- MACRO CAL, an optimizing assembly language.
- FORTRAN V, 1966 ANSI standard with extensions for process control.
- FORTRAN VI, a block optimizing compiler.
- COBOL WITH ISAM Indexed Sequential Access Method, conforms to American Standard X3.23 – 1974.

Interactive extended BASIC is also available in single user and multi-user versions.

MEMORY ACCESS CONTROLLER (MAC) OPTION

MAC translates a 20-bit program address into a 20-bit physical address. In addition, it protects memory from illegal accesses, prevents write access to blocks of memory, rejects instruction execution from blocks of memory, and detects invalid memory addresses. MAC provides relocation, segmentation, and protection on 256-byte boundaries.

Sixteen 32-bit segmentation registers specify the relocation starting address, length of the block in 256-byte "pages" and protection status. Each program segment can be from 256 bytes to 64K bytes long.

Large systems that support multiprogramming and run under OS/32MT require the MAC option. It is also used in shared memory configurations for mapping a processor's logical addresses into shared memory space.

SPECIFICATIONS

Technology
  Processor TTL-MSI and LSI
  ROM Bipolar (60ns access time)
Data Word Length – 8, 16, 32, 64 bits
Instruction Word Length – 16, 32, 48 bits
Number of Basic Instructions – 132
Additional instructions for
Extended Branch Mnemonics – 58
  Single/Double Precision Floating Point – 34
  High Speed Data Handling Option – 3
Fixed Point Arithmetic – 2's Complement
Floating Point Arithmetic – sign/magnitude (6 or 14 hexadecimal digit precision)

Hardware Accumulators
  32 Fixed Point – Two sets of 16 each – 32 bits
  8 Single Precision Floating Point – 64 bits
  8 Double Precision Floating Point – 64 bits
Hardware Index Registers – 15 per Register Set
Address Modes – Direct, Indexed, Relative, and Immediate

PROCESSOR OPTIONS

Memory Access Controller
Memory Parity Check & Generate
Power Fail/Auto Restart
DMA Buffer
Universal Clock
Extended Memory Selector Channel
Turnkey Console
Binary Display Console
Hexadecimal Display Console
Local Memory Bank Interface (required for each additional 256KB of memory)
High Speed Data Handling Option
Single and Double Precision Floating Point Option (hardware)

INPUT/OUTPUT

Input/Output Modes – Programmed transfers, 26K to 150K 8 or 16-bit transfers/second
– Block transfers up to 360K bytes/second
– Auto driver channel, 70K bytes or halfwords/second
– 7 DMA ports available
– DMA Transfer – 2M bytes/second
Priority Interrupts – One level and identification of up to 1024 hardware priority interrupts with automatic device identification and vectoring.
*Interrupt Overhead Time – 6.5 μs
Normal Interrupt Latency Time – 4.0μs
Hardware I/O Timeout – 14μs timeout on all micro processor operations to the I/O system, thereby ensuring that the processor will not lock-up should a module stall or otherwise fail to respond.

*Overhead time to allow servicing interrupts, including device identification, status recognition, and PSW swapping.

MAIN MEMORY

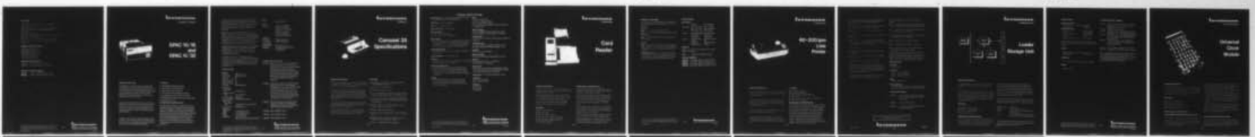
Word Length – 16 bits (17 with parity option)
Organization – 64 KB and two types of 32KB modules available on single 15" plug-in boards
Cycle Time – 32KB – 750 or 1000 nanoseconds
  64KB – 1000 nanoseconds
Maximum Memory Size – 1,048,576 Bytes

ENVIRONMENTAL

Temperature: 0-50°C
Humidity: 0-90% (non-condensing)
Vibration: 0-55 Hz at 1.25 G's
Storage Temperature: [illegible]

AD-A052 359 RAYTHEON CO WAYLAND MASS ADVANCED DEVELOPMENT LAB F/G 17/9
R AND D EQUIPMENT INFORMATION REPORT: TRACKING AND SIGNIFICANCE--ETC(U)
NOV 77 T I BOAK, A J JAGODNIK, R B MARSHALL F19628-77-C-0148
UNCLASSIFIED ER77-4283 AFGL-TR-77-0259 NL

2 OF 2
AD
A052359



END
DATE
FILMED
5-78
DDC

PACKAGING

19-in. RETMA chassis, 14-in. high, 28-in. deep (48.3 cm. x 34.6 cm. x 69.1 cm.)
15-in. x 15-in. (38.1 cm. x 38.1 cm.) printed circuit boards with 1/4-in. (0.64 cm.) aluminum stiffeners arranged horizontally
Printed circuit backpanel
Pin to dual contact receptacle connections with locating pins
Dual chassis with 16-15" board slots
Processor occupies 3 boards, each memory module 1 board
Power 115 or 230 VAC ± 10%, 50 or 60 Hertz, 16 amps max. (at 115 volts)

OPTIONAL DISPLAY CONSOLE

36 Binary Indicating LED's
9 Hexadecimal Display Matrices
16 Hexadecimal Character Keys
10 Function Select Keys
5 Function Indicating LED's
1-3 Position Keylock Switch

RELATED DOCUMENTATION

29-399R03	Model 7/32 Reference Manual
29-400R00	Model 7/32 Processor User's Manual

INTERDATA PRODUCT NUMBERS

M73-032	7/32-CII with 64KB, 750nsec memory
M73-033	7/32-CII with 64KB, 1000nsec memory

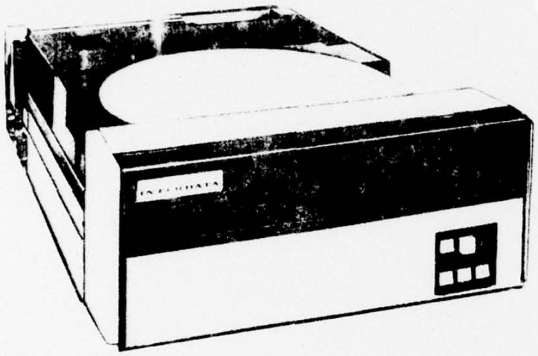
The information contained herein is intended to be a general description and is subject to change with product enhancement.

Printed in U.S.A.

INTERDATA
A UNIT OF
PERKIN-ELMER DATA SYSTEMS
2 CRESCENT PLACE ● OCEANPORT, NEW JERSEY 07757

INTERDATA

PERIPHERALS SERIES W



DPAC 10/16 and DPAC 10/32

PRODUCT DESCRIPTION

The DPAC10 series of disc systems are completely self-contained random access bulk storage subsystems for use with INTERDATA's wide range of processor systems. The basic DPAC10 system provides a formatted storage capacity of over 10.02 million bytes. By adding up to three DPAC10/E Disc System Expansions, the system can be expanded to over 40.1 million bytes.

Used with Interdata 16- and 32-bit computer systems, the DPAC10/16 and DPAC10/32 systems provide mass storage facilities for a wide variety of small and medium scale requirements; complete software file management and data recovery capabilities are provided by standard Interdata operating systems.

Included in the basic system is the Direct Memory Access controller, a disc controller capable of controlling four disc drive units, cables, a 10-megabyte removable cartridge disc drive, and a formatted removable disc cartridge.

FEATURES

- 10,027,008 Byte Formatted Capacity
- Expandable to 40.108 Million Bytes
- 33-millisecond Average Seek Time
- 12.5-millisecond Average Rotational Latency
- 312,5000 Bytes-per-Second Disc Transfer Rate
- 2,000,000 Bytes-per-Second System Throughput Rates
- Hardware Write Protect

OPERATIONAL CHARACTERISTICS

The DPAC10 series systems use an IBM-type 5440, two-recording-surface disc cartridge with a nominal formatted capacity of 5 megabytes. The additional 5 megabytes are recorded on a fixed two recording-surface disc with both discs using a common spindle and head-actuating mechanism. The exceptionally fast and accurate head-positioning mechanism allows radial positioning of the four heads to any one of the 408 available cylinders. The average positioning time is 33 milliseconds, track-to-track time is 10 milliseconds, and a maximum full-stroke time is 60 milliseconds.

The rotation speed is a highly accurate 2400 RPM using a DC brushless motor that is impervious to line power variations. Data transfers at a rate of 312,500 bytes per second with the Direct Memory Access controller capable of autonomous data transfer rates of up to 2 million bytes per second.

Data security and integrity are assured by the use of extensive error detection logic in the controller, a constant revolution DC-powered spindle, disc cleaning brush assembly to minimize particle contamination, head-positioning limiter and power loss protect circuits causing the heads to be fully retracted, and a positive cartridge interlock mechanism preventing removal of the cartridge until fully stopped.

The Direct Memory Access and disc controllers are standard Interdata 15-inch by 15-inch (38.1 x 38.1 cm) printed circuit boards that can be inserted into any standard Interdata chassis. The disc drive unit can be mounted in a standard Interdata cabinet.

The DPAC10/16 and DPAC10/32 systems are fully supported by the wide range of standard Interdata software. Included is the BOSS operating system, the powerful multitasking OS/16MT2 and OS/32MT operating systems, and higher level languages such as FORTRAN, COBOL, BASIC and CORAL 66.

SPECIFICATIONS

Capacity	
Bit Density	2200 bits per inch
Track Density	200 tracks per inch
Cylinder	408
Tracks per Cylinder	4
Bytes per Sector	256
Sectors per Track	24
Formatted	10,027,008 bytes
Access Time	
Average Access Time	33 milliseconds
Average Rotational Delay	12.5 milliseconds
Rotational Speed	2400 RPM
Disc Transfer Rate	312,500 byte per second
DPAC System Throughput Rate	2,000,000 bytes per second (max.)
Disc Cartridge	IBM 5440 type
Dimensions	
Height	7.75 inches (19.7 cm)
Width	17.6 inches (44.7 cm)
Depth	22 inches (55.9 cm)
Controllers	Two 15 x 15 inch (38.1 x 38.1 cm) printed circuit boards
Weight	
	95 pounds (43 kg)
Power	
Disc Drive	120 VAC, 3.3 amperes, 47-63 Hz single phase, starting current 12 amperes. 230 VAC, 1.65 amperes, 47-63 Hz single phase, starting current 6 amperes.
Controllers	
DPAC10/16	+5VDC, 6.8 amperes
DPAC10/32	+5VDC, 10.5 amperes
Environmental	
Temperature	19° to 38° C operating
Humidity	10 to 80% Relative Humidity, no condensation

INTERDATA Product Numbers

M46-645 DPAC10/16 Disc System. A complete storage subsystem consisting of a Direct Memory Access controller, a subsystem controller capable of controlling up to four disc drives, a 10 million byte formatted capacity disc drive, and a formatted cartridge. Subsystem throughput rate to 2 million bytes per second, device transfer rate to 312,500 bytes per second. For use with DMA capable 16-bit Interdata processors.

M46-646 DPAC10/32 Disc System. A complete storage subsystem consisting of a Direct Memory Access controller, a subsystem controller capable of controlling up to four disc drives, a 10 million byte formatted capacity disc drive, and formatted cartridge. Subsystem throughput rate to 2 million bytes per second, device transfer rate to 312,500 bytes per second. For use with 32-bit Interdata processors.

M46-647 DPAC10/E Disc System Expansion. Allows expansion of DPAC10 and DPAC20 Disc Systems with a 10-million byte capacity disc drive, formatted cartridge and cables.

M46-650 Same as M46-645. 50Hz.

M46-651 Same as M46-646. 50Hz.

M46-652 Same as M46-647. 50Hz.

The information contained herein is intended to be a general description and is subject to change with product enhancement.

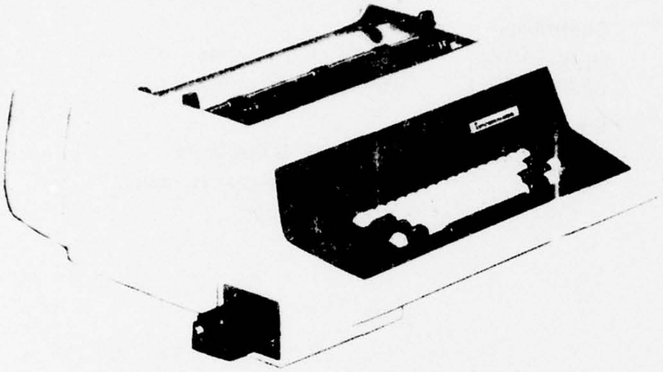
INTERDATA
A UNIT OF
PERKIN ELMER DATA SYSTEMS
2 CRESCENT PLACE • OCEANPORT, NEW JERSEY 07757

Printed in U.S.A. April, 1977

INTERDATA

TERMINALS

Carousel 35 Specifications



PRODUCT DESCRIPTION

The Carousel 35 is a serial impact computer terminal with print quality rivaling the Selectric*, multiple copy capability, and provision for a 120 cps paper tape reader.

The Carousel 35 provides the capabilities of a console I/O device high speed paper tape reader and a low speed line printer in a single, highly cost effective package. Carousel 35 printing throughput is 40 to 70 percent the speed of the M46-204 60-200 LPM Line Printer for typical computer output formats.

Special forms handling features, Electronic Format Control, and total legibility make the Carousel especially useful in business and financial environments.

The Carousel 35 print cup gives fully formed character images for clean, crisp copy up to a million impressions. Cups are easily interchanged for a variety of fonts.

FEATURES

- Printing Speed – 40 to 70 percent throughput of the M46-204 60-200 LPM Line Printer.
- Quality Print – Rivals Selectric*.
- Multiple Copies – Clear up to 12
- Forms handling flexibility – Pin-feed continuous, friction-feed rolls, cut sheets, ledger cards, stock certificates, etc.
- Electronic format control – Adjustable left margin, 1/48-inch vernier for any line spacing, programmable horizontal and vertical tabs.
- Wide Font Selection – Operator changeable, machine readable. Courier 72, APL, OCRA.
- Plotting – 4800 points per square inch.
- Cartridge Ribbons – Red/black fabric, carbon film, black fabric.
- Operator oriented – No-mess cartridge, typewriter-like keyboard, Communications Control Panel.
- Service – Interdata/Perkin-Elmer worldwide.

CAROUSEL 35 SPECIFICATIONS

Speed, Printing – 30 characters per second; Spacing – to 200 cps; Paper Movement – 5 inches per second

Print Type – Fully formed characters

Fonts – Courier 72, APL, OCR-A

Number of Columns – 80,132 optional

Printing Characters
64 ASCII (Upper Case) Standard
96 ASCII (Upper/Lower Case) Optional

Paper Handling
Friction feed platen path for cut or single-part continuous forms. Unique, patent pending friction-feed mechanism permits handling forms from 3 inches to 5 inches in width with less than 0.5 inches lateral movement over any length form.

Paper Handling Options
Line-printer-like tractor driven path for heavy, multipart continuous forms.

Horizontal Spacing – 10 characters per inch

Keyboard
Typewriter-like layout with a convenient, optional, numeric pad for financial and scientific applications.

Electronic Format Control
Optional addressable horizontal and vertical positioning. Adjustable left margin. Forms-length/top-of-form control for forms up to 127 lines in length. Program controlled vertical vernier with 1/48-inch resolution for non-standard line spacing.

Plotting
Optional plotting mode permits 1/100-inch horizontal resolution and 1/48-inch vertical resolution plotting. Unique Fastplot mode permits plotting throughput of 60 increments per second.

Copies
1 to 6-part forms standard
12-part form capability optional
Stencil cutting capability optional
No operator adjustments required to switch from single to multi-part forms.

Ribbon Cartridges
Operator changeable, no-mess cartridges. Black fabric, single-strike carbon film, and red/black fabric available.

Vertical Spacing
6 lines per inch with 1/48-inch fine vernier for accurate forms positioning.

Red/Black Printing
Option permits operator mounting of red/black ribbon cartridges in addition to standard single-color ribbons.

Interfaces
20 Ma current loop, 1320 Band.

Paper Tape Reader
Optional optical paper tape reader operates at 120 cps.

Buffering
Communication Line Input – 128 characters
Keyboard Buffer – 32 characters

Space Jump™
Microprocessor automatically converts sequences of spaces in the input buffer to horizontal tab operations. Spacing at rates to 200 characters per second is possible.

Acoustic Cover – For extra quiet operation, optional.

Input Power
120 VAC, 60 Hz Standard
240 VAC, 50 Hz Optional

Dimensions
32 inches wide
24 inches deep
9 inches high

The information contained herein is intended to be a general description and is subject to change with product enhancement.

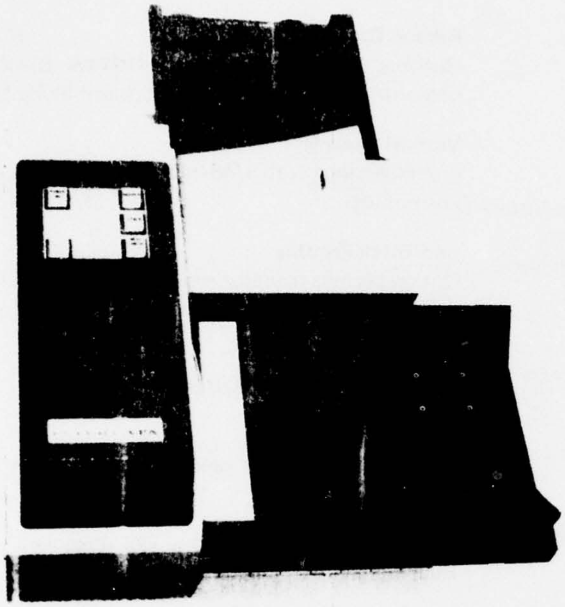
INTERDATA
A UNIT OF
PERKIN-ELMER DATA SYSTEMS
2 CRESCENT PLACE ● OCEANPORT, NEW JERSEY 07757
5761015

Printed in U.S.A.

INTERDATA

PERIPHERALS

Card Reader



PRODUCT DESCRIPTION

The Interdata Card Reader is an extremely compact, lightweight, free-standing unit featuring optional hardware Hollerith-to-ASCII conversion.

As a highly desirable program and data entry medium, the card reader lends itself to complete program development and large diverse data entry requirements. Simplicity in design permits easy operator interface and monitoring. All switch controls are easily accessible and provide a full range of functions/indications.

FEATURES

- 400 or 1000 CPM readers
- 80 Column Cards
- 1500 Card Hopper capability
- 1500 Card Stacker capability (1000 CPM)
- 500 Card Stacker capability (400 CPM)
- Optical Read Mechanism

OPERATIONAL CHARACTERISTICS

The Card Reader has a Fiber Optic Read Station, card transport drum feed assembly, and several data verification techniques which preclude loss of data.

The card read rate is either 400 or 1000 cards per minute with up to 1500 card capacity for both input hopper and output stacker. Throughout the read operation, "Light Test", "Dark Test", and card motion tests are continuously performed to verify the operation of the card reader.

A Card Feed Command causes the card to move against the card transport drums elastomer surface, selecting the card for reading. This propels the card past a throat knife assembly, and into the read station.

As the card passes the fiber bundles and light distribution block, it is wiped to eliminate dust in the read station.

Data Read is available in two bytes per column. The first byte occupies the top six rows of a column; the second byte occupies the bottom six rows of that column.

CONTROL – INDICATORS

The Card Reader incorporates all the control and indicator functions necessary for complete operator control and monitoring.

Indicators

Power – A green indicator denotes power ON.

Hopper – A red indicator identifies an empty input hopper or feed failure.

Stack – A red indicator identifies a full output stacker or a stack failure.

Controls

Reset – The control resets electronics and conditions the reader to a READY or NOT READY condition depending upon previous condition. The switch contains a white indicator.

Read Check/Lamp Test – A red illuminated momentary contact switch which identifies a read station failure, or possible data error; when depressed tests all front panel indicators if the reader is not in the ready condition.

SPECIFICATIONS

Environmental

	Operating	Non Operating
Temperature	+50° F to + 100° F Maximum gradient 0.2° F per minute	–30° F to 150° F Maximum gradient 20° F per hour
Humidity	30% to 80% relative humidity; no condensation	5% to 95% relative humidity, no condensation
Altitude	1000 feet below sea level 10,000 feet above sea level	1000 feet below sea level 35,000 feet above sea level

Dimensions

Reader – 17 in. high by 14 in. wide by 18.5 in. deep
Weight – 55 pounds

Power 115/230 VAC, 60/50 Hz, 5/3 Amp

PRODUCT NUMBERS

M46-238	Card Reader, 400 cpm 115 volts AC 60 Hz
M46-239	Card Reader, 400 cpm 230 volts AC 50 Hz
M46-244	Card Reader, 1000 cpm 115 volts AC 60 Hz
M46-245	Card Reader, 1000 cpm 230 volts AC 50 Hz

The information contained herein is intended to be a general description and is subject to change with product enhancement.

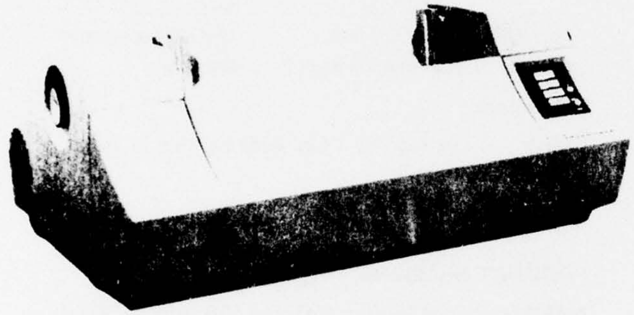
INTERDATA
Subsidiary of PERKIN-ELMER • Oceanport New Jersey 07757

Printed in U.S.A.

INTERDATA®

PERIPHERALS

60–200 lpm Line Printer



PRODUCT DESCRIPTION

Interdata's 60 to 200 lpm Line Printer is a low-cost serial printing line printer that is ideally suited for light-to-medium printing requirements.

Printing at a serial rate of 165 characters per second, 10 characters per inch, and up to 132 characters per line, the line printing speed ranges from 60 full lines per minute to 200 short lines per minute.

The printer is interfaced to the Interdata processor multiplexor bus via a character parallel interface.

The Line Printer has pin feed handling for forms up to 14-3/8 inches wide, a full line buffer, and a two-channel vertical format control for top-of-form and vertical tabulation.

FEATURES

- 60 to 200 lines per minute
- Pin-feed forms handling
- Variable width forms to 14-3/8 inches
- Full line buffer
- Two-channel vertical format top-of-form control
- Completely self contained

OPERATIONAL CHARACTERISTICS

The Line Printer is completely self-contained and includes the mechanical and electromechanical components, control logic, character pattern generator, line buffer and power supply. Matrix printing is used. The print head consists of print solenoids mounted on a carrier. Printing is accomplished through selective pulsing of the print wires as the print head moves from left to right across the character space and the print line. The solenoids are activated independently up to five times for any one character. This technique provides enough force to produce excellent legibility using six part form paper.

Four control switches and three indicator lights are housed on the operator's panel. The control switches are:

- Stop-Start Switch – Supplies power to the printer unit in the start position when switch is illuminated.
- Top of Form Switch – Used for manual slewing to top of form.
- Select Switch – Used to select the printer after turning on power.
- Forms Override – Provides operator override on the paper out switch, allowing the operator to complete the form being printed before changing paper.

Indicator lights are:

- Hardware Alarm – Indicates head carrier has exceeded right margin.
- Paper Out – Indicates out of paper or a paper handling malfunction.
- Multiple Purpose – Used in special applications.

In addition to the alarm indicators, an audio alarm is activated for the out of paper condition, paper handling malfunction, and if the head carrier exceeds the right hand margin. The audio alarm also sounds when a Bell Code is received under program control.

The parallel interface to the Serial Printer is contained on a standard Interdata 7-inch printed circuit board.

The printer recognizes six non-printing commands as follows:

- CR – Causes the printer buffer to print, causes the carrier to return, and causes the form to be advanced one line.
- FF – Causes the form to be advanced to the first punch in Channel 1 of the form feed tape.
- VT – Causes the form to be advanced to the first punch in Channel 2 of the form feed tape.
- LF – Causes the form to be advanced one line.
- BELL – Activates the audio alarm for 2 seconds.
- SO – Causes the buffer to print elongated characters (66 characters per line instead of 132).

The 60-200 lpm Line Printer is fully supported by a wide range of standard Interdata software. This includes the powerful OS16/MT2 and OS/32 MT operating systems, the BOSS-Plus operating system, and higher level language implementation such as COBOL, FORTRAN, and BASIC.

SPECIFICATIONS

Print Speed

60 to 200 LPM

Dimensions

Width – 27½ inches (69.8 cm)
Height – 11¼ inches (29 cm)
Depth – 19¼ inches (48.9 cm)

Weight

155 pounds (70.3 Kg)

Power Requirements

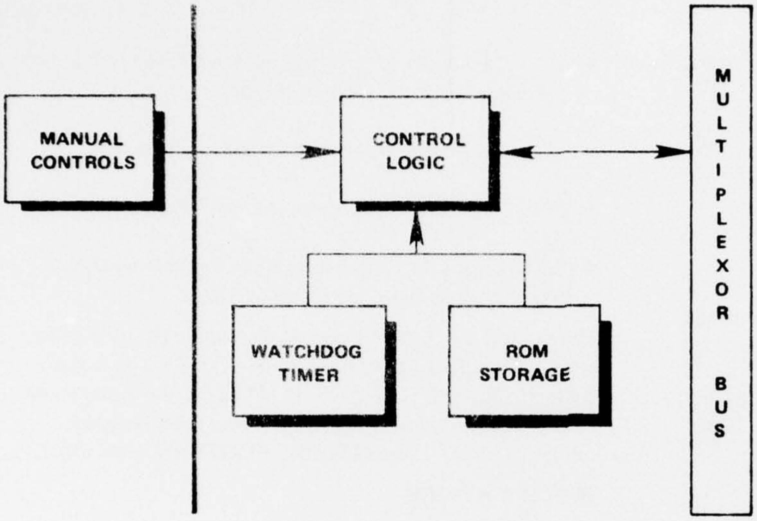
120 VAC, 15 amperes (maximum), single phase, 60 Hz.
230 VAC, 7.5 amperes (maximum), single phase, 50 Hz.

Interdata Product Numbers

M46-204	Fully buffered line printer, 60 to 200 lpm, 132 columns, 64 character set, includes external cable.
M46-205	Same as M46-204. 50 Hz.
M46-202	Line Printer Interface and internal cable for 60-200 lpm Printer.

Information in this bulletin is not an explicit specification and is subject to change at any time.

INTERDATA
Subsidiary of PERKIN ELMER • Oceanport New Jersey 07757
Printed in U.S.A.



Loader Storage Unit

PRODUCT DESCRIPTION

The Loader Storage Unit is used primarily as an automatic initialization/restart device for remote or unattended systems.

The controller is designed to give the user the utmost flexibility in initialization program format and watchdog timer sensing. The controller is capable of housing 2,048 bytes of PROM in 128-byte segments. The incorporated watchdog timer may be strapped for a timing interval from 16 milliseconds to 256 milliseconds.

FEATURES

- Unattended Program Reload
- Standard Operating System Loaders
- Custom Program Parameters
- Program Loop Detection

OPERATIONAL CHARACTERISTICS

The unit is contained on one printed circuit board, has a fixed device assignment, and plugs into the standard Interdata Multiplexor Bus. Unlike most device controllers, the Loader Storage Unit does not generate an interrupt.

The watchdog timer is used to signal the initialization sequence. Under normal operating conditions the timer is reset by a software generated output command prior to the preset time out delay. Should the program fail or enter an infinite loop, the timer will time-out and the restart sequence may be initiated. The unit may also be used to restart upon restoration of power after a power failure.

Three switches are provided for manual control of the unit. The switches are bracket-mounted and connect to the Loader Storage Unit via a 10-foot cable. The switches are as follows:

DESIGNATION	FUNCTION
ON/OFF	Inhibits addressing the unit.
ENAB	Allows initialization of the system upon watchdog time-out.
INIT	Unconditionally initializes the system if the ON/OFF switch is in the ON position.

Several predefined loader programs are available for both 16- and 32-bit systems. These loaders allow automatic loading of operating systems from disc I/O devices.

SPECIFICATIONS

STORAGE CAPACITY

Loader Storage Unit Controller	2,048 Bytes
Storage Module	128 Bytes

WATCHDOG TIMER

Minimum Interval	16 milliseconds
Maximum Interval	256 milliseconds*
Selectable Increment	16 milliseconds
Accuracy	± 5%
Standard Strapped Interval	96 milliseconds

*A user may customize the timer over the range 10 milliseconds to 10 seconds by changing two RC networks in the controller.

POWER

Loader Storage Unit Controller	1.0 Amp @ + 5V
Storage Module	0.1 Amp @ + 5V

DIMENSIONS

7 X 15 inch (17.8 X 38.1 cm) Printed Circuit Board

WEIGHT

2 Pounds (.91 Kg)

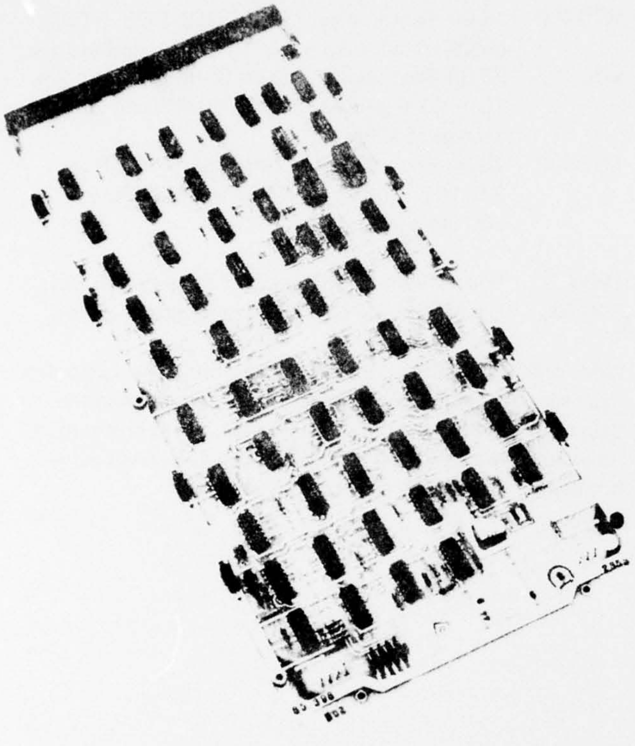
INTERDATA PRODUCT NUMBERS

M70-104	Loader Storage Unit Controller with hardware watchdog timer and IC sockets for up to 16 each M70-105 Storage Modules.
M70-105	128 Byte Storage Modules.
M70-106	16-Bit Boot Loader. Loads BOSS, DOS, RTOS, or OS/16 MT 1 from 2.5, 10 or 40 megabyte disc.
M70-107	32-Bit Boot Loader. Loads OS/32 MT from 2.5, 10 or 40 megabyte disc and MSM80 and 300 Storage Modules.
M70-108	16-Bit Boot Loader. Loads OS/16 MT2 from 2.5, 10 or 40 megabyte disc and MSM80 and 300 Storage Modules.
S90-402	Loader Storage Unit Support Program (16 Bit).
S90-403	Loader Storage Unit Support Program (32 Bit).

When ordering customer designed storage modules, customers must supply a binary paper tape or a listing specifying the data to be written into the LSU PROMs. This is prepared in accordance with program S90-402 or S90-403. The LSU is not used with the 5/16 or 6/16 processors.

The information contained herein is intended to be a general description and is subject to change with product enhancement.

INTERDATA
A UNIT OF
PERKIN-ELMER DATA SYSTEMS
2 CRESCENT PLACE ● OCEANPORT, NEW JERSEY 07757



Universal Clock Module

PRODUCT DESCRIPTION

The Universal Clock Module is a versatile timer consisting of two clock devices, a highly accurate crystal controlled Precision Interval Clock (PIC) and an AC Line Frequency Derived Clock (LFC). Each of the clocks is completely independent of the other for maximum convenience.

FEATURES

- Crystal controlled accuracy
- External synchronization
- Programmable load and sense
- Programmable resolution and interval
- Two Independent Clocks

OPERATIONAL CHARACTERISTICS

PROGRAMMABLE PRECISION INTERVAL CLOCK

The Precision Interval Clock is dynamically variable through program control. It provides timer controlled processor interrupts and a program accessible counter giving resolutions of 1, 10, and 100 microseconds as well as 1 millisecond through an interval range of 2^{12}.

The master time base for the PIC is provided by a one megahertz crystal oscillator which can be disabled to allow substitution by an external master time base oscillator.

1. Resolution and Interval data are retained in the PIC input buffer, where it resides until new data is supplied.
2. Timing is started upon command and continues until the interval is reached, at which time a program interrupt is generated. The timing operation automatically continues in a cyclic manner. The user may dynamically alter the interval any time before the completion of the present interval.
3. The PIC incorporates an Output Buffer to allow Interval Counter interrogation without disturbing its operation.

The PIC interrupt circuitry can be disabled under program control. This action does not inhibit the timing operation or the ability to sense the interval count.

AC LINE FREQUENCY DERIVED CLOCK

The line frequency is derived directly from the AC power line and is presented as a clock rate equal to twice the line frequency. This clock has no set-up procedure other than to enable, disable or disarm the interrupt circuit.

SPECIFICATIONS

UNIVERSAL CLOCK MODULE

Accuracy:	PIC – ± .01% Crystal controlled Oscillator
Resolution:	(PIC) 1 μs, 10 μs, 100 μs, 1 ms
Interval:	PIC 1 μs to 4,095 μs, 10 μs to 40,950 μs, 100 μs to 409,500 μs, 1 ms to 4,095 ms. (corresponding to resolution) LFC 8.33 ms on 60 Hz line 10 ms on 50 Hz line
Program Control:	PIC – Control to Clock Command – Disable, Disarm, Enable, Start Status – Overflow Write Data Byte 1, 2 – Resolution and Interval Count PIC – Output to Processor Read Data 1, 2 – Current Interval Count LFC – Command – Disable, Enable, Disarm
Power Requirements:	1.75 AMP @ + 5 V 12 VAC 10 ma
Dimensions:	7" X 15" (17.8 X 38.1 cm) Printed Circuit Board
Weight:	1.0 Pounds (.5 kg)

INTERDATA PRODUCT NUMBER

M48-000	Universal Clock Module

The information contained herein is intended to be a general description and is subject to change with product enhancement.

Printed in U.S.A.

INTERDATA
A UNIT OF
PERKIN-ELMER DATA SYSTEMS
2 CRESCENT PLACE ● OCEANPORT, NEW JERSEY 07757

INTERDATA

Product Bulletin

TELETYPEWRITER CURRENT LOOP INTERFACE

- **110 or 1200 Baud**
- **7 x 15 Inch Printed Circuit Board**
- **Console Control or Local Terminal Interface**
- **8 Level Code (USASCII)**
- **Character Buffered**
- **High Reliability**

GENERAL DESCRIPTION

The INTERDATA Teletypewriter Current Loop Interface is a compact, efficient and highly reliable interface that may be used to interface a local 20 milliampere terminal device for use as a control console or terminal device to the CPU.

The interface incorporates complete control and status control facilities to insure data transmission integrity. The 20 milliampere current required by the local device is provided in the interface.

Two interfaces are available. One operates at 110 baud (10 characters per second), the other at 1200 baud (110 characters per second). The interface automatically appends and strips the start and stop bits required for asynchronous 11 unit transmission coding.

OPERATIONAL CHARACTERISTICS

The interface is designed for connection to an INTERDATA processor multiplexor bus and is contained on a single 7 x 15 printed circuit board.

Complete program control is exercised via command and status bytes. The commands available allow control over interrupt status, reading and writing data and the paper tape facilities when connected to an automatic send-receive unit. Status monitoring allows sensing character over-write attempts, line spacing exceeding a single character time, device not available and device busy.

Data is transmitted and received in a bit serial stream with a bit time space appended to start the character and two stop bits (two mark bit frames) appended at the end of the eight character bits. A serial bit shift register is used to interface to the line, this register is connected to a parallel buffer register to provide full character buffering. Data presented to, or received from the processor multiplexor bus is interfaced via the character buffer register to allow character parallel transfers.

Timing control utilizes RC controlled multivibrator clocking base circuits and control circuitry to clock the data bits.

The timing frame of the clock circuits allows a 9.09 millisecond bit period for 110 baud data framing. The full character frame, including the start and stop bits is 100 milliseconds.

The timing frame of the clock circuits for 1200 baud operation provides an 833 microsecond bit period and a full character frame of 9.163 milliseconds.

The interface address may be strapped to any desired legal address in the range 1 to 256 to allow system design flexibility.

SPECIFICATIONS

Data Transmission Rate – 110 Baud (10 characters per second)
1200 Baud (110 characters per second)

Transmission Mode – Bit Serial

Electrical Interface – 20 Milliamperes Current Loop (current provided by interface internal power)

Character Code – 8 Level (bit) character

Start Bits – 1

Stop Bits – 2

Parity – Even

Commands – Disable Interrupt, Enable Interrupt, Disarm Interrupt, Unblock, Block, Write Data, Read Data

Status – Overflow, Break, Busy, Examine, Device Unavailable

Power Requirements – 1.6 Amperes @ +5 VDC

Operating Environment – 0 to 50°C
10 to 90% Humidity (without condensation)

Dimensions – 7 x 15 inch printed circuit board

Weight – 1 pound

INTERDATA Product Numbers

M48-~~010~~	ASR Model 33/35 Interface with internal cable.
M46-107	1200 Baud Local Current Loop Interface with internal cable.

Sales and Service Offices:

New York
2460 Lemoine Avenue
Fort Lee, New Jersey 07024
(212) 736-8540

Rochester
2509 Browncroft Blvd.
Rochester, New York 14625
(716) 385-3083

Boston
60 Hickory Drive
Waltham, Mass. 02154
(617) 890-0557

Philadelphia
6 Valley Forge Executive Mall
676 Swedesford Road
Suite 232
Wayne, Pa. 19087
(215) 687-2511

Pittsburgh
615 Iron City Drive
Pittsburgh, Pa. 15205
(412) 922-8904

Washington
1800 No. Kent St., Suite 101
Rosslyn, Virginia 22209
(703) 525-4806

Orlando
7200 Lake Ellenor Drive, Suite 142
Orlando, Florida 32809
(305) 851-6962

Atlanta
1925 Century Lane
Suite 4
Atlanta, Georgia 30345
(404) 321-5151

Chicago
605 E. Algonquin Road
Arlington Heights, Illinois 60005
(312) 437-5120

Detroit
20100 Civic Center Drive, Suite 320
Southfield, Mich. 48076
(313) 356-5515

Kansas City
Clover Leaf Office Building No. 1
6811 W. 63rd St., Suite 204
Shawnee Mission, Kansas 66202
(913) 384-1606

Dayton
Financial South Office Park
5335 Far Hills Avenue
Kettering, Ohio 45429
(513) 434-4193

Los Angeles
888 No. Sepulveda Blvd. Suite 666
El Segundo, California 90245
(213) 640-0451

San Diego
7841 Balboa Avenue
San Diego, California 92111
(714) 565-0602

Phoenix
1801 So. Jen Tilly Lane, Suite C-6
Tempe, Arizona 85281
(602) 968-2477

San Francisco
Ponderosa Office Building
3080 Olcott St., Suite 125A
Santa Clara, California 95051
(408) 249-5540

Denver
1660 So. Albion, Suite 225
Denver, Colorado 80222
(303) 758-0474

Seattle
400 108th Ave., N.E., Suite 607
Bellevue, Washington 98004
(206) 455-0680

Dallas
13771 No. Central Expressway, Suite 624
Richardson, Texas 75231
(214) 234-8880

Houston
6610 Harwin Drive, Suite 230
Houston, Texas 77036
(713) 783-3060

Tulsa
Ramada Professional Building, Suite 16
4580 E. 50th St.
Tulsa, Oklahoma 74135
(918) 664-1260

Mexico City
Informatica Nacional, S. A.
Insurgentes Sur No. 933, 13er Piso
Mexico 18, D. F., Mexico
543-15-42

Vancouver
1401 W. Broadway, Suite 20
Vancouver 9, BC V6H 1H6
Canada
(604) 736-2801

Toronto
6486 Viscount Road
Mississauga, Ontario, Canada L4V 1H3
(416) 677-8990

Ottawa
2841 Riverside Dr., Suite 303
Ottawa, Ontario K1V 8N4
Canada
(613) 731-0116

Montreal
9675 Cote de Liesse
Suite 102
Dorval, Quebec, H9P 1A3
Canada
(514) 636-6410

Duesseldorf
4 Duesseldorf
Bonnerstrasse 7-11
West Germany
(0211) 79-2090

Munich
8032 Grafelfing/Munich
Waldstrabe 3A
(854-20-34-38)

London
Arundel Road
Uxbridge, Middlesex, England
Uxbridge 52441

Canberra
Block 9 Section 27
27-29 Collie Street
Fyshwick ACT 2609
Canberra, Australia
95-9073

Melbourne
153 Park Street
South Melbourne, VIC 3205
South Melbourne, Australia
69-7786

Sydney
92 Chandos Street
St. Leonards
Sydney, Australia 2065
439-8400

South Africa
Protea PNI (Pty.) Ltd.
7 Newton Street
Wemmer, Johannesburg
Transvaal, South Africa
838-8351

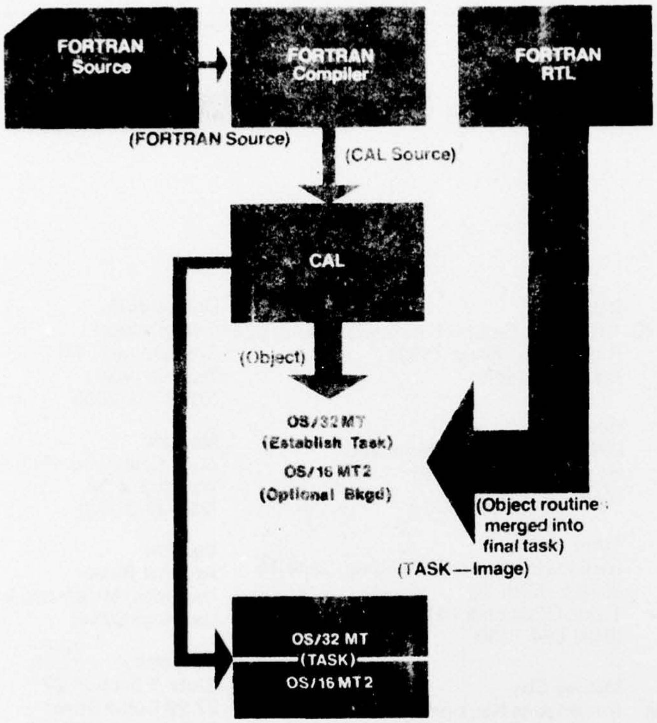
Tokyo
Kyokuto Boeki Kaisha, Ltd.
15-4 2 Chome
Higashi-Nihonbashi
Chuo-Ku, Tokyo
862-6488

INTERDATA 2 Crescent Place, Oceanport, New Jersey 07757 (201) 229-4040 Printed in U.S.A. No. 59574-S

The information contained herein is intended to be a general product description and shall not be utilized as an explicit specification for such product.

INTERDATA

SOFTWARE



FORTRAN V Level 1

PRODUCT DESCRIPTION

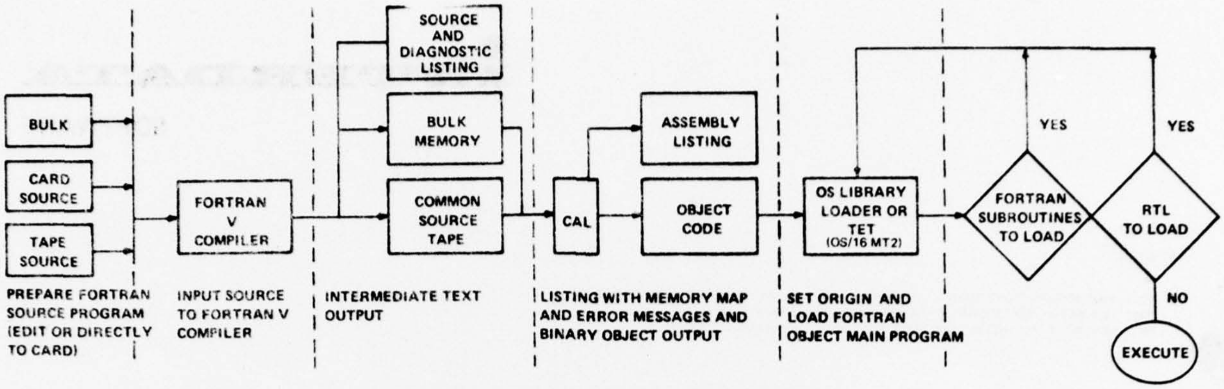
FORTRAN V, Level 1 language extensions provide a rich FORTRAN vocabulary as well as support for the ISA real-time extensions and special applications.

FORTRAN V Compiler supports all the features of ANSI FORTRAN Standard X3.9-1966. In addition, the Purdue Instrument Society of America (ISA) real-time extensions are supported, including bit and byte manipulation features. The compiler permits device-independent I/O on a wide range of media.

FORTRAN Run Time Library Routines are used by the FORTRAN object programs at execution time to implement I/O operations, complex and double-precision arithmetic, and intrinsic and external functions provided in the ANSI Standard. These routines are reentrant. Thus a single copy of the library can concurrently service multiple FORTRAN object programs.

FEATURES

- Data structures and program sizes to 1 megabyte
- General language extensions make compiler a superset of ANSI (USASI) Standard X3.9–1966
- Special language extensions for systems development and process control (Purdue/ISA Features)
- Object code targeted for either 16-bit or 32-bit processors
- In-line assembly coding capability
- Run-time trace capability
- I/O device independence
- Error location pin pointing
- Source listing and object code options
- Reentrant Run-Time FORTRAN Library
- Conditional compilations
- Random file access


SOURCE AND DIAGNOSTIC LISTING
BULK
CARD SOURCE
TAPE SOURCE
FORTRAN V COMPILER
BULK MEMORY
COMMON SOURCE TAPE
CAL
ASSEMBLY LISTING
OBJECT CODE
OS LIBRARY LOADER OR TET (OS/16 MT2)
YES
YES
FORTRAN SUBROUTINES TO LOAD
RTL TO LOAD
NO
EXECUTE
PREPARE FORTRAN SOURCE PROGRAM (EDIT OR DIRECTLY TO CARD)
INPUT SOURCE TO FORTRAN V COMPILER
INTERMEDIATE TEXT OUTPUT
LISTING WITH MEMORY MAP AND ERROR MESSAGES AND BINARY OBJECT OUTPUT
SET ORIGIN AND LOAD FORTRAN OBJECT MAIN PROGRAM

FORTRAN V PROGRAM OPERATION

LARGE PROGRAMS AND DATA STRUCTURES

The compiler generates an intermediate text which is in turn assembled by the Common Assembly Language (CAL) to produce machine-level code for both Interdata 16-bit and 32-bit processors. When a 32-bit processor is the target machine, the compiler allows program and data sizes to the limit of configured memory – 1 megabyte.

GENERAL LANGUAGE EXTENSIONS

FORTRAN V, Level 1 general language extensions provide the following:

- Mixed mode arithmetic
- *Hollerith constants and string declarations*
- Data statements that allow array initialization and implied Do-Loops
- Multiple entry subroutines
- Error and End-of-File Returns from Read/Write statements

SPECIAL LANGUAGE EXTENSIONS – SYSTEM DEVELOPMENT

These extensions together with the general extensions provide high level language systems software development capability. The list of extensions includes in-line assembly code capability, random file accessing, and conditional compilations.

SPECIAL LANGUAGE EXTENSIONS – Process Control (Purdue/ISA)

The FORTRAN V compiler supports a subset of the extensions agreed upon by the Instrument Society of America/Purdue University. These extensions are implemented as external functions for bit and byte manipulations on integers as follows:

Bit Manipulations

- Inclusive Or
- Logical Product
- Logical Complement
- Exclusive Or
- Logical Shift
- Bit Test

Byte Manipulations

- Byte Load
- Byte Store
- Byte Clear
- Byte Compliment

SYSTEM REQUIREMENTS – THE FORTRAN PROGRAMMING SYSTEM

The FORTRAN Programming System has the following key elements:

1. FORTRAN V, Level 1 Compiler.
2. The Run-Time Library which includes all FORTRAN Intrinsic and External functions plus additional I/O and arithmetic subroutines.
3. *One of the applicable operating systems:*
   - Basic Operating System (BOSS)
   - Disc Operating System (DOS)
   - Real Time Operating System (RTOS)
   - OS/16 MT2 (Multi-Tasking)
   - OS/32 ST (Serial Tasking)
   - OS/32 MT (Multi-Tasking)
4. The OS Library Loader, a command-driven loader for ease of loading compiled programs when utilizing Interdata operating systems with Task Establisher Task (TET).
5. The CAL Assembler, which translates the common source text produced by the compiler into machine code for either the 16-bit or 32-bit series processors.

MINIMUM SYSTEM CONFIGURATIONS

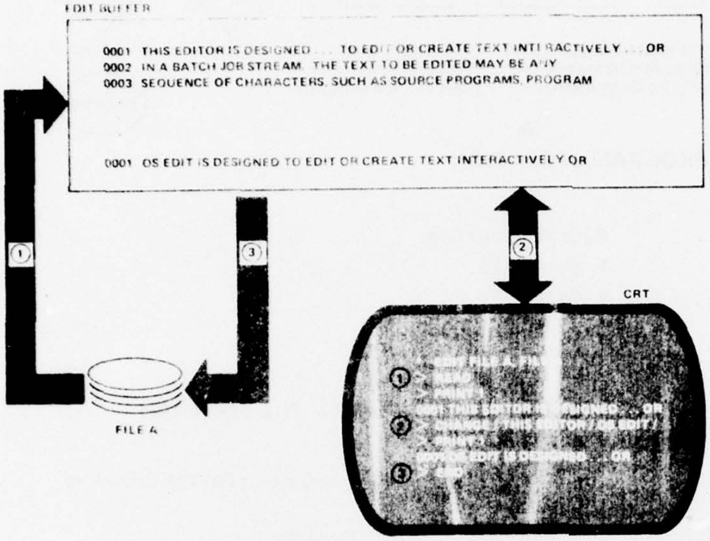
Processor Model	BOSS	OS/16MT2	OS/32 ST	OS/32 MT
6/16 7/16	X X	X X		
7/32 8/32			X X	X X
Minimum Memory Size	40KB	64KB	96KB	96KB

Information in this bulletin is not an explicit specification and is subject to change at any time.

INTERDATA
Subsidiary of PERKIN ELMER • Oceanport New Jersey 07757
FEB 76-I-004
Printed in U.S.A.

INTERDATA

SOFTWARE



```
0001  THIS EDITOR IS DESIGNED . . . TO EDIT OR CREATE TEXT INTERACTIVELY . . . OR
0002  IN A BATCH JOB STREAM. THE TEXT TO BE EDITED MAY BE ANY
0003  SEQUENCE OF CHARACTERS, SUCH AS SOURCE PROGRAMS, PROGRAM

0001  OS EDIT IS DESIGNED TO EDIT OR CREATE TEXT INTERACTIVELY OR
```

OS EDIT

GENERAL DESCRIPTION

OS EDIT is a text editor which satisfies the needs of the user who has simple editing requirements or who requires complex editing facilities.

OS EDIT is designed to edit or create text interactively or in a batch job stream. The text to be edited may be any sequence of characters, such as source programs, program data, or documents. The format of the input text may be ASCII or binary giving the user the capability to modify source files or object modules by simply specifying the input data mode.

The user may examine, change, or delete characters according to context or line reference numbers. Text is read into an area of memory called the Edit Buffer (EB). The EB is set to a default size of 1 KB and may be extended by operator command according to the memory that is available.

SYSTEM REQUIREMENTS

An Interdata 16-bit or 32-bit processor with the minimum configuration required for OS/16 MT or OS/32 MT.

INTERDATA PRODUCT NUMBER

03-063   OS Edit

FEATURES

OS EDIT is a comprehensive text editor that provides the following features:

- String Search and Replacement Capability
- File Manipulation Capability
- Flexible Text Handling – Text may be operated upon either on a character basis or a line basis as desired.
- Ease of Use – English command oriented mnemonics that facilitate learning and use of edit commands.
- Operational Flexibility – OS EDIT may be run in Batch mode or interactively from a terminal such as a Carousel or CRT.
- Multiple Text Formats – Text may consist of ASCII characters or binary data from any input device.
- Compatibility Across Operating Systems – OS EDIT may be run under the control of OS/16 MT or OS/32 MT.

The information contained herein is intended to be a general description and is subject to change with product enhancement.

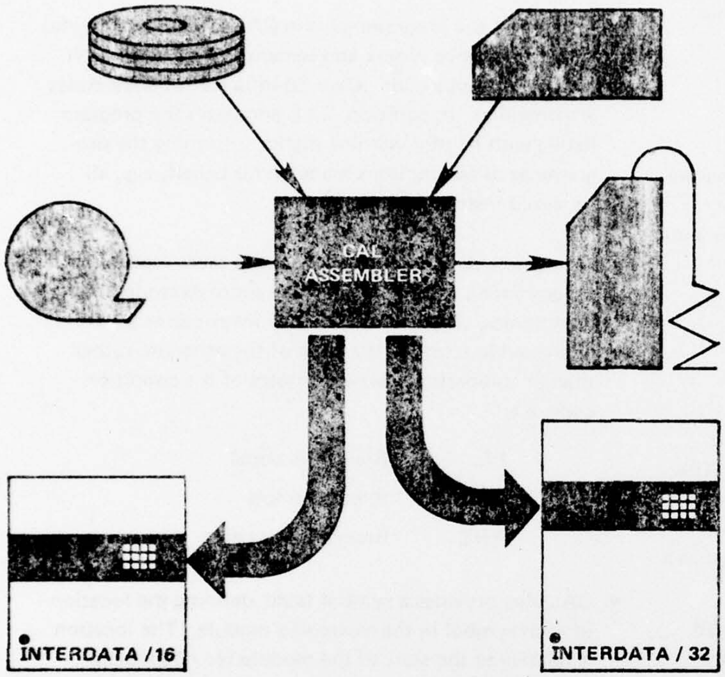
Printed in U.S.A.

INTERDATA
Subsidiary of PERKIN-ELMER • Oceanport New Jersey 07757
9761036

INTERDATA

SOFTWARE

CAL Assembler



PRODUCT DESCRIPTION

Interdata's Common Assembly Language (CAL) brings the power of the Interdata computer systems to the programmer's fingertips. Easy to use, CAL provides many facilities to simplify the programming task and to efficiently implement programs.

The same language is provided on both the 16-Bit and 32-Bit systems. A common language mode is available so that programs can be run on either type of machine, a facility of particular importance at installations with a mixture of Interdata 16-Bit and 32-Bit systems. Retraining costs are virtually eliminated.

The assembler provides powerful features which encourage efficient, flexible user programs: optimization of machine instructions ensures the use of the shortest practical form of each machine operation; conditional assembly features enable extremely adaptable source programs; data structure definition and manipulation mechanisms provide mnemonic access to complex structures; and, source library facilities provide convenient standardization across a family of programs.

FEATURES

- Machine Instruction Optimization
- 16- and 32-Bit Operation; Cross-assembling
- Common Language Mode
- Debugging Aids
- Data Structure Definitions
- Conditional Assembly
- Library Facilities
- High-Level Language Integration
- Small 16-Bit Version

MACHINE INSTRUCTION OPTIMIZATION

As an option, the CAL assembler optimizes individual instructions to their shortest equivalent form. In general, RX3 instructions are reduced to the RX1 or RX2 form, immediate instructions are "squeezed" to their shortest form, and branches are shortened, where admissible. Thus:

```
AI   R1,1      becomes    AIS    R1,1
AI   R1,24     becomes    AHI    R1,24
BNE  LOOP      becomes    BNES   LOOP
```

The effect of machine instruction optimization is to generate most efficient object code without violating one-for-one translation of source statements. Squeezed source statements are flagged in the program listing.

16- AND 32-BIT OPERATION

The CAL assembler operates on either Interdata 16-Bit or 32-Bit Computer Systems, under the control of an appropriate operating system (OS/32MT on the 32-Bit machines; OS/16MT2 or BOSS-PLUS on the 16-Bit.) The options available in these environments are identical and either version of the assembler may be used to generate 16- or 32-Bit programs. CAL operates as an assembler or as a cross-assembler.

CAL generates object code for all Interdata 16- and 32-Bit computers. The object code is processed by the appropriate linkage editor (TET/32, TET/16 or Library Loader) to generate loadable program modules. The linkage editor is responsible for locating code, which is usually relative within its module. Facilities are provided, however, for code to be absolutely located by the assembler.

COMMON SOURCE LANGUAGE

In addition to the machine-specific facilities offered by CAL, a Common language mode is provided which allows a single source program to run on either machine.

The use of this facility is particularly appropriate for utility and special purpose programs required to run on both machines. Programs written in this manner cannot normally make optimum use of the facilities of either type of machine; the small amount of run time efficiency lost is often outweighed by the large savings in implementation and maintenance costs.

CAL itself is written in Common language mode, as are a number of Interdata supplied utilities, such as OS COPY and OS Source Updater.

DEBUGGING AIDS

The ultimate test of any program language system is its effectiveness in creating fully tested, operational programs. The CAL assembler provides the following aids that make this process as efficient as possible:

- The assembler provides free-format input within each line of source. Input lines may be dedicated to comments and each source statement may be individually commented. A comprehensive listing is provided, with the generated object code adjacent to the source code.

- To protect the programmer, the CAL assembler provides a comprehensive syntax and semantic error check as it assembles source code. Over 30 informative error codes are provided. In addition, CAL annotates the program listing with helpful warning marks, informing the programmer of assumptions made on his behalf, e.g., all squeezed instructions are marked.

- A further aid, which makes programs more readable and programming more natural, is the set of extended branch mnemonics. These enable branch instructions to be expressed in terms of the logic of the program, rather than as comparisons between states of the condition code, e.g.:

BE	Branch on equal
BL	Branch on low
BNZ	Branch on not zero

- CAL also provides a symbol table, defining the location of each symbol in the assembled module. The location is relative to the start of the module for relocatable symbols and absolute for absolutely located symbols. A cross-reference chart is available as an option, providing an ordered list of every reference to each symbol.

DATA STRUCTURE FACILITIES

CAL provides two mechanisms for defining data structures, which can then be referenced mnemonically. These are 'COMN' for defining FORTRAN compatible common blocks, and 'STRUC' for defining local data structures.

References to a common block are resolved by the Task Establisher (TET/32, TET/16) which is responsible for actual allocation of space for common. Local data structures are normally used for indexed access into data areas within a program.

CONDITIONAL ASSEMBLY

CAL provides a flexible set of conditional assembly facilities. By embedding suitable control statements into a CAL program, flexibility can be built into a source program. By setting the control variables appropriately, a program can be adapted during assembly to the environment in which it is to run. An example of sophisticated use of this facility is the system generation mechanism of OS/16MT2, which provides system level adaptability entirely through the use of conditional assembly.

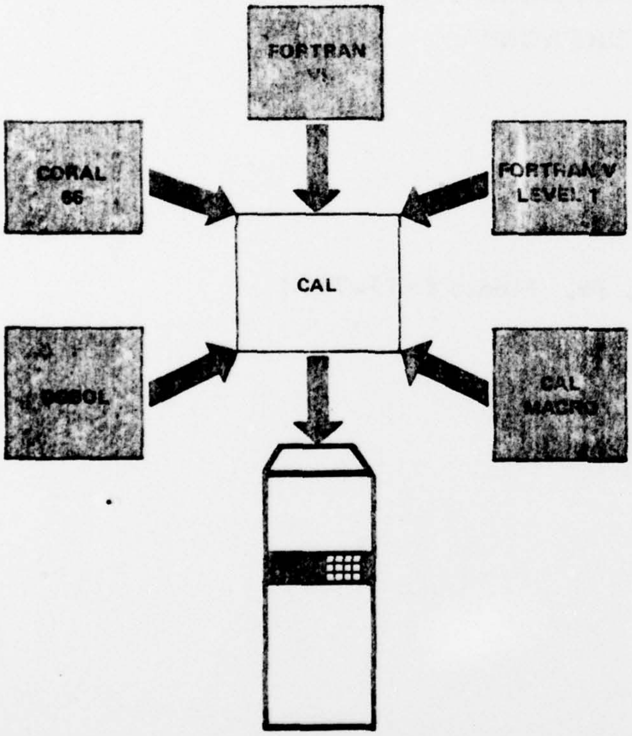
LIBRARY FACILITY

Through the 'COPY' facility, CAL provides a source library feature. Identified source modules can be incorporated into a program by means of a COPY operation in the program source.

A typical use of this facility occurs when a number of programs are making use of similar data structures. For example, the source of OS/32MT includes a COPY file of data structures defining the SVC parameters blocks, UDL, TCB, etc.

HIGHER LEVEL LANGUAGES

As well as providing assembly facilities, CAL is also used as the second pass for a number of Interdata compilers, viz: FORTRAN VI and COBOL on the 32-Bit lines, FORTRAN V, CORAL 66 and CAL MACRO on both the 16- and 32-Bit lines.



SMALL 16-BIT VERSION

A 16-Bit version of CAL (known as CAL/16 or CAL/16D) is provided with OS/16MT2. This version of the assembler supports only 16-Bit source code, and is intended for use in situations where processing resources are limited.

CAL/16 is entirely memory bound, requiring 14KB plus symbol table space to perform assemblies. CAL/16D makes use of disc files for symbol tables and requires 16.5KB of memory beyond requirements of the OS/16MT2 operating system.

OPERATIONAL REQUIREMENTS

CAL requires 25.5KB on a 32-Bit processor and 24.5KB on a 16-Bit processor, beyond the requirements of the operating system. In addition, memory must be available for symbol table space. In typical operational usage, symbol tables vary in size from virtually zero to 16KB and more. Most assemblies can be performed in 8KB of symbol table space.

PROGRAM NUMBERS

S90-204	CAL Assembler
03-101	CAL/16 and CAL/16D Assemblers

CAL is provided as a component of many Interdata software packages, including:

S90-006	OS/32MT
S90-201	FORTRAN V Level I
S90-212	32-Bit FORTRAN V
S90-213	FORTRAN VI
S90-214	COBOL
S90-205	CAL Macro

CAL/16 and CAL16/D are provided as components of

S90-010	OS/16MT2

APPENDIX G

"CURSOR DATA TRANSFER IN THE DISPLAY DATA INTERFACE"

(A. J. Jagodnik, Jr. Memo #AJJ-76A)

RAYTHEON

FORM 10-0557 (9-65) BOND

DIVISION	EQUIPMENT	Contract No.	F19628-77-C-0148
Operation	EDL		
Department	ADL - Wayland	Distribution	cc
To	File	File No.	-
From	A. J. Jagodnik, Jr.	Memo No.	AJJ-76 A
Subject	Cursor Data Transfer in the Display Data Interface	Date	30 December 1977
		Revised:	5 January 1978

Classification Unclassified

The Display Data Interface (DDI) is a circuit card first developed under contract number DNA 001-75-C-0050 for the Liquid Water Content Analyzer. The DDI provides for various types of data transfers between a Scan Converter/Refresh Memory (SCRM) and an Interdata 7/32 minicomputer. The particular type of transfer of concern in this memo is known as "Cursor Data Entry", in which the operator first turns the CURSOR switch to ON for one or more display channels, then locates a cursor on a display screen by means of a CURSOR POSITION Trackball, and finally pushes a SEND DATA button for the desired display channel. The DDI then requires up to 16.7 milliseconds to acquire the data after which it generates an interrupt to the 7/32. In an interrupt service routine, the 7/32 subsequently reads the following data from the DDI: 1) a two-bit code indicating which display channel the data came from; 2) a four-bit color code from the channel's image memory at the cursor coordinates, and 3) the X (9 bits) and Y (8 bits) cursor coordinates. The DDI, as developed for the Liquid Water Content Analyzer is fully documented in section 4.6 (hardware) and Appendix D, memo AJJ-21 (programming) of the final report, DNA 4130F.

A second DDI of similar design was built for the third SCRM under contract F19628-76-C-0101, however, this DDI differs from the earlier design because SCRM-3 has a time-multiplexed memory bus feature which permits simultaneous operations in different display channels. In terms of programming, the major difference is in the status byte which has a "Display Channel Not Available" bit for each of the four channels. The second DDI is documented in the Equipment Information Report for contract F19628-76-C-0101, in sections 2.1.1 (General), 3.1.10 (Operation of Control Panel), 4.7 (Detailed Circuit Description), Appendix C (memo AJJ-21 which still applies except for differences with the LWCA) and Appendix D (Cursor Data Entry Test Program).

A problem in programming the cursor data entry sequence in the LWCA was encountered by Lloyd Perry of ERT. This multiple entry effect was traced to a hardware design problem which was corrected by implementing minor changes in the LWCA DDI in July, 1976. A short machine language test program was written to verify proper operation of the cursor data entry sequence; it is this program which appears in Appendix D of the Equipment Information Report for contract F19628-76-C-0101. The same hardware changes were performed on the SCRM-3 DDI; however, the test program was never run on this system because a 7/32 was not available at the time. Because the program in Appendix D is short and can be keyed-in through the 7/32's hex display panel, and since it contains comments useful in understanding the program to implement cursor data entry, running this test program on the SCRM-3 and 7/32 (which are now part of the Echo Tracking and Significance Estimator) would have two beneficial results:

1. Verification that the Cursor Data Entry hardware operates properly.

2. Familiarization of the programmer with the machine-language implementation of the Cursor Data Entry interrupt service routine.

Several minor changes to the program are needed. First, change the data at address 2002 to 00CD so the SCRMI won't be put in its test mode. Next, change 2006 to 008C since the DDI device code is different from that in LWCA. Finally, put 2020 at 01E8 in the interrupt service pointer table.

A. J. Jagodnik, Jr.
Advanced Electronic Techniques
Wayland Box M9, x5171

AJJ/lld

cc: F. P. Eggleton
R. B. Marshall
M. J. Young

K. Glover (3)
G. Armstrong
K. Banis
A. Bishop
} AFGL